

**A FRAMEWORK FOR CONCURRENT DESIGN AND ROUTE PLANNING
OPTIMIZATION OF UNMANNED AERIAL VEHICLE BASED URBAN
DELIVERY SYSTEMS**

A Dissertation
Presented to
The Academic Faculty

By

Younghoon Choi

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology

August 2019

Copyright © Younghoon Choi 2019

**A FRAMEWORK FOR CONCURRENT DESIGN AND ROUTE PLANNING
OPTIMIZATION OF UNMANNED AERIAL VEHICLE BASED URBAN
DELIVERY SYSTEMS**

Approved by:

Professor Dimitri N. Mavris,
Advisor, Committee Chair
School of Aerospace Engineering
Georgia Institute of Technology

Professor Daniel Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Bradford E. Robertson
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Kelly Griendling
Center of Innovation for Aerospace
*Georgia Department of Economic
Development*

Dr. Zohaib T. Mian
Loon (Alphabet Inc.)

Date Approved: May 16, 2019

ACKNOWLEDGEMENTS

I would like to show my thanks to my committee, Dr. Dimitri N. Mavris, Dr. Daniel Schrage, Dr. Bradford E. Robertson, Dr. Kelly Griendling, and Dr. Zohaib T. Mian. Without their advice, guidance, and support, this thesis would not have been possible. Thank you for taking the time and the effort to keep me on track throughout the process. I would like to especially thank my advisor, Dr. Dimitri N. Mavris, the director of the Aerospace Systems Design Laboratory. I have learned numberless lessons about life as well as engineering. I am thankful for all of his advice, guidance and support.

I would also like to give a special thank you to Dr. Bradford E. Robertson and Dr. Youngjun Choi from ASDL. They have spent a substantial amount of their time to discuss my immature ideas to make them clearer and more concrete throughout the process.

Finally, I would like to express my deepest gratitude to my parents, wife, and kids for their love, support, and patience.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	iii
LIST OF TABLES	ix
LIST OF FIGURES	xi
LIST OF ACRONYMSxviii
SUMMARY	xxi
Chapter 1: Introduction	1
1.1 Unmanned Aircraft Systems (UAS)	1
1.2 Delivery Systems based on a sUAS	4
1.3 Aircraft Sizing and Synthesis Process	5
1.3.1 Traditional Aircraft Sizing and Synthesis Process	6
1.3.2 Aircraft Sizing and Synthesis Process for a fleet-level mission	7
1.4 Challenges for the operation of sUAS-based delivery systems	12
1.5 Research Motivation	13
1.6 Research Scope	22
1.7 Research Question and Objective	25
Chapter 2: Background	27

2.1	Fixed Point Iteration (FPI) Method	27
2.1.1	Fixed Point Iteration Theory	28
2.1.2	Fixed Point Iteration for a System of Nonlinear Equations	32
2.1.3	Fixed Point Iteration Method	34
2.2	Vehicle Routing Problem (VRP)	35
2.2.1	Graph Theory	35
2.2.2	Combinatorial Optimization Problem	36
2.2.3	Integer Programming (IP)	37
2.2.4	Complexity	39
2.2.5	Existing Traveling Salesman Problem (TSP)	41
2.2.6	Existing Vehicle Routing Problem (VRP) Models	44
2.3	Summary	52
Chapter 3: Mathematical modeling of a small UAS-based Delivery System . . .		53
3.1	Development of Hypothesis	53
3.2	Optimization Model for an Endurance-Constrained MTRPTW	56
3.3	Experiments	60
3.3.1	Experiment 1: Comparison of Optimization Models	61
3.3.2	Experiment 2: Sensitivity Analysis	63
3.4	Conclusions	67
Chapter 4: Construction of a Two-Layered Urban Flight Network		68
4.1	Development of Hypothesis	69
4.2	Creating a Framework for a Two-Layered Urban Flight Network	73

4.2.1	Step 1: Creating an Urban Model	74
4.2.2	Step 2: Capturing Urban Obstacles and Restricted Airspace	75
4.2.3	Step 3: Constructing a Low-level Urban Flight Network	77
4.2.4	Step 4: Building a High-level Urban Flight Network	79
4.3	Experiments	81
4.3.1	Experiment 3: Validation of a Two-Layered Urban Flight Network	83
4.3.2	Experiment 4: Sensitivity Analysis	84
4.4	Conclusions	88
 Chapter 5: Implementation of a Sizing and Synthesis Process for small fixed-wing VTOL UAVs		 90
5.1	Development of Hypothesis	91
5.2	Aircraft Sizing and Synthesis Process for a VTOL UAV	92
5.2.1	Mission Profile	93
5.2.2	Constraint Analysis	94
5.2.3	Weight Analysis	95
5.3	Experiments	101
5.3.1	Experiment 5: Single-Stop Delivery Mission	103
5.3.2	Experiment 6: Two-Stop Delivery Mission	105
5.4	Conclusions	107
 Chapter 6: A Framework for Concurrent UAV Design and Vehicle Routing Problems for Urban Delivery		 109
6.1	Development of Hypothesis	110

6.2	Creating a Framework for Concurrent UAV Design and Routing Problems for urban delivery	113
6.2.1	Optimization Model for an Energy-Constrained MTRVPTW	115
6.3	Experiments	118
6.3.1	Experiment 7: Convergence Analysis	121
6.3.2	Experiment 8: Sensitivity Analysis	128
6.4	Conclusions	134
Chapter 7: Conclusions and Future work		136
7.1	Conclusions	136
7.2	Future Work	141
7.3	Contributions	142
Appendices		144
Appendix A: The Vehicle Routing Problems: Applications and Approaches . . .		145
A.1	UAS-based Package Delivery: an Extended Savings Algorithm	145
A.1.1	Key Rules of the Extended Saving Algorithm	145
A.1.2	The Algorithm Structure of the Extended Savings Algorithm	153
A.1.3	Merge Strategies: Sequential vs. Parallel	153
A.2	Coverage Path Planning: a Column Generation Method	157
A.2.1	Problem Modeling	157
A.2.2	Optimization Model	161
A.2.3	Column Generation of the RBECOM for CPP Problems	169

A.3	Multi-UAS Path-Planning for a Large-scale Disjoint Disaster Management: Mathematical Formulations	179
A.3.1	Introduction of Disaster Management Missions	179
A.3.2	A Framework for a UAS-based Wildfire Scanning Mission	180
Appendix B:	Regulatory Requirements for sUAS	193
References	204

LIST OF TABLES

1.1	Notional UAS vehicle classification and categorization [1]	3
1.2	Summary of literatures for the concurrent aircraft design and vehicle operation problem	18
1.3	Input and output of the UAV design module and vehicle routing module	20
3.1	Summary of total cost optimized schedules of the 16 scenarios	66
4.1	Comparison of the five groups of path-planning methods	71
4.2	Results of the effects of flight networks	83
4.3	Cost coefficients by different vehicle specifications	84
4.4	Results of the effects of vehicle properties and time windows with first demand set	86
5.1	Assumptions for sizing and synthesis process	102
5.2	Mission segments of a single-stop delivery	103
5.3	Mission profile of a single-stop delivery	104
5.4	Mission segments of a two-stop delivery	105
5.5	Mission profile of a two-stop delivery	106
6.1	Summary of initial value analysis of the objective function	122
6.2	Summary of initial value analysis of the flight time	122

6.3	Summary of initial value analysis of the vehicle cost	122
6.4	Effect of initial conditions on the solution of both the endurance-constrained MTVRPTW and the framework for concurrent vehicle design and routing with the first demand scenario	126
6.5	Effect of initial conditions on the solution of both the endurance-constrained MTVRPTW and the framework for concurrent vehicle design and routing with the second demand scenario	127
6.6	Summary of sensitivity analysis of the first demand scenario	129
6.7	Summary of sensitivity analysis of the second demand scenario	132
B.1	Summary of small unmanned aircraft rule (Part 107) [170]	196
B.2	Comparison of model aircraft, section 333 exemptions, and proposed regu- lations under the small UAS notice of proposed rulemaking [7]	201

LIST OF FIGURES

1.1	Total UAS forecast 2015 - 2035 [1]	2
1.2	Operating altitudes, endurance, and weight of common UAS [7]	3
1.3	Notional delivery system	4
1.4	The developing sVTOLs for a package delivery mission [12, 13, 14, 15, 16, 17]	5
1.5	The Concept of Operations (ConOps) and mission profile of a delivery mission	7
1.6	Nam's sizing method [23]	8
1.7	Gundlach's sizing method [24]	8
1.8	Example of a delivery mission requiring multiple UAVs	9
1.9	The first possible optimal ConOps for a fleet-level delivery mission including seven mission points	10
1.10	The second possible optimal ConOps for a fleet-level delivery mission including seven mission points	10
1.11	Notional sizing and synthesis process with a fleet-level mission	11
1.12	Two approaches to optimizing a plan for a fleet-level mission	12
1.13	Diagram of the integrated transportation system model [30]	14
1.14	Integrated transportation system design optimization with simulated annealing [30]	14
1.15	Sequential decomposition schematic [29]	16

1.16	Problem formulation and decomposition with uncertainty [29, 31]	16
1.17	Decomposition framework [32]	17
1.18	Simultaneous Aircraft Sizing and Allocation Decomposition [33]	17
1.19	Simplified structure of the MDO models	19
1.20	Interaction between the UAV design and vehicle routing modules	20
1.21	Hi-level framework for concurrent UAV design and routing for urban delivery systems	22
1.22	Structure of chapters of this thesis	26
2.1	FPI for a system consisting of two nonlinear equations	34
2.2	Modeling a flight network by a graph	36
2.3	Example of a traveling salesman problem	42
2.4	Taxonomy of the VRP literature [25]	45
2.5	Notional example of a Vehicle Routing Problem (VRP)	46
2.6	Notional example of a Vehicle Routing Problem with Time Windows (VRPTW)	48
2.7	Notional example of a Multi-Trip Vehicle Routing Problem with Time Windows (MTVRPTW)	51
3.1	Time line for reusability constraint	58
3.2	Time line for endurance constraint	58
3.3	Total cost optimized solution of a VRPTW model (baseline)	61
3.4	Total cost optimized solution of an endurance-constrained VRPTW model	63
3.5	Total cost optimized solution of an endurance-constrained MTVRPTW model	64
3.6	Summary of simulation results	65

4.1	The framework for building a two-layered urban flight network	74
4.2	Mission area: San Diego, CA (Google Image)	74
4.3	The effect of operating altitude on capturing urban obstacles	76
4.4	Restricted airspace and operating space with polygonal obstacles	76
4.5	Procedure of creating a low-level urban flight network	78
4.6	Effect of an artificial boundary on a road map	78
4.7	Voronoi paths with polygonal obstacles	80
4.8	Effects of flight networks on a solution	83
4.9	Two demand scenarios	85
4.10	Demand distributions of two demand scenarios	85
5.1	Process of the developed fixed-wing VTOL UAV sizing module	93
5.2	Constraint analysis graph consisting of cruise, climb, and stall	95
5.3	Regression models of propeller performance: thrust, velocity, and rpm . . .	98
5.4	Regression models of propeller performance: P_{shaft} , velocity, and rpm . . .	98
5.5	Regression models of motor and propeller performance: throttle setting and P_{in}	99
5.6	Regression models of motor and propeller performance: P_{in} and rpm	99
5.7	Regression models of motor and propeller performance: P_{in} and thrust . . .	99
5.8	Mission profile of a single-stop delivery	103
5.9	Initial design point of a single-stop mission	104
5.10	Final design point of a single-stop mission	104
5.11	Mission profile of a two-stop delivery	105
5.12	Initial design point of a two-stop mission	106

5.13	Final design point of a two-stop mission	107
6.1	A notional example of a solution of the vehicle routing module	111
6.2	A dynamic mission profile	111
6.3	Convergence criteria of the FPI method for concurrent UAV design and vehicle routing problems	112
6.4	Structure of a FPI method for a concurrent UAV design and vehicle routing problem	112
6.5	Flow chart of the integrated framework	114
6.6	The two demand scenario	120
6.7	Results of convergence analysis of two demand scenarios	123
6.8	Results of sensitivity analysis of the first demand scenario	130
6.9	Results of sensitivity analysis of the first second scenario	133
A.1	An example of merging two trips	147
A.2	The effect of time window on the order of visiting customers	148
A.2	The effect of time window on the order of visiting customers (cont.)	149
A.3	Maximum hovering rule	150
A.4	Maximum endurance rule	151
A.5	Multi-trip rule	152
A.6	Comparison of two merge strategies	156
A.7	An illustration of a coverage path planing problem	157
A.8	Preferred routes for the coverage path planning problems	157
A.9	A mission profile of a UAV for an imagery mission	158
A.10	Block angular structure of a constraint matrix	165

A.11 A simple network topology consisting of 6 nodes and 10 edges	170
A.12 Change in the domain of routes during the column generation process . . .	171
A.13 Column generation framework	171
A.14 Shapes of constraint matrix at the i -th and the $(i+1)$ -th iterations of column generation.	175
A.15 Notional decomposition of the large-scale scanning mission	181
A.16 The effect of a line-sweep direction on creating a coverage path	183
A.17 The effect of turning performance on turns of a Dubins' vehicle	184
A.18 The modeling of mission areas by a graph	186
A.19 The effect of vehicle routing formulations on a ConOps	187
B.1 Airspace classification	194
B.2 Amazon's plan to use airspace	194

LIST OF ACRONYMS

AGL Above Ground Level

AOI Area Of Interest

ATC Analytical Target Cascading

ATSP Asymmetric Traveling Salesman Problem

BLDC BrushLess DC

BVLOS Beyond Visual Line-Of-Sight

CCO Convex Combinatorial Optimization

CFD Computational fluid Dynamics

CO Collaborative Optimization

ConOps Concept of Operations

CPP Coverage Path Planning

CVRP Capacitated Vehicle Routing Problem

DoD Department of Defense

DoE Design of Experiments

ESC Electronic Speed Controller

ESP Euclidean Shortest Path

EV RPTW Electric Vehicle Routing Problem with Time Windows

FAA Federal Aviation Administration

FARs Federal Aviation Regulations

FM Figure of Merit

FPI Fixed Point Iteration

GVRP Green Vehicle Routing Problem

HALE High-Altitude Long-Endurance

IP Integer Programming

LASE Low-Altitude Short-Endurance

LCO Linear Combinatorial Optimization

LiDAR Light Detection And Ranging

LP Linear Programming

MDO Multi-Disciplinary Optimization

MILP Mixed-Integer Linear Programming

MINLP Mixed-Integer Non-Linear Programming

MIP Mixed Integer Programming

MTVRPTW Multi-Trip Vehicle Routing Problem with Time Windows

NP Nondeterministic Polynomial time

OR Operations Research

PWM Pulse Width Modulation

RO Research Objective

rpm revolutions per minute

RQ Research Question

RRT Rapidly-exploring Random Tree

STSP Symmetric Traveling Salesman Problem

sUAS small Unmanned Aircraft Systems

sUAVs small Unmanned Aerial Vehicles

TFRs Temporary Flight Restrictions

TSP Traveling Salesman Problem

UA Unmanned Aircraft

UAS Unmanned Aircraft Systems

UAV Unmanned Aerial Vehicle

VLOS Visual Line-Of-Sight

VRP Vehicle Routing Problem

VRPTW Vehicle Routing Problem with Time Windows

VTOL Vertical Take-Off and Landing

SUMMARY

With the emergence of new technologies for small Unmanned Aircraft Systems (sUAS), such as lightweight sensors and high-efficiency batteries, the operation of small Unmanned Aerial Vehicles (sUAVs) has expanded from military use to commercial use. A promising commercial application of sUAS is package delivery because of its potential to reduce acquisition and operating costs of the last-mile delivery system while enabling new services such as same-day delivery. Furthermore, in urban areas, sUAVs can deliver packages to customers without negatively affecting street traffic.

This thesis addresses an extended sizing and synthesis process that considers the performance of a sUAS in its total cost optimized routing to size the vehicles for sUAS-based delivery systems. This problem is called the concurrent aircraft design and routing problem. Based on decomposition approaches, this problem can be divided into three parts: the sizing, the route planning, and the integration of the two. However, the existing methods have mainly focused either UAV design or optimization of operations of UAVs. Although a concurrent UAV design and routing problem is addressed, only simple routing problems are studied without considering the obstructed environment like an urban area.

To fully address the concurrent aircraft design and routing problem for sUAS-based delivery systems, this thesis presents a novel modular framework including all three parts of this problem: the UAV design module, the UAV routing module, and the integration method. First, for the UAV routing module, this thesis presents an endurance-constrained Multi-Trip Vehicle Routing Problem with time windows (MTVRPTW) optimization model that is an extension of the MTVRPTW optimization model. The MTVRPTW model builds a vehicle's schedule including a reuse plan for an on-demand delivery system. However, the MTVRPTW model does not consider the property of sUAV's limited endurance. To alleviate the limitation of the MTVRPTW model, the endurance-constrained MTVRPTW model employs maximum endurance constraints that trace flight time of each vehicle and restrict

flight time to vehicle's maximum endurance. Moreover, to address the urban environment with the optimization model, this thesis presents a framework for creating a two-layered urban flight network as an input graph of a Vehicle Routing Problem (VRP) optimization model. The urban flight network is built by feeding airborne Light Detection And Ranging (LiDAR) sensor data into an algorithm that uses a Voronoi diagram to create collision-free paths. By integrating the endurance-constrained MTRPTW model with the two-layered urban flight network, vehicle's schedule for sUAS-based urban delivery is created.

Second, for the UAV design module, a component-based sizing and synthesis process for small fixed-wing VTOL UAVs is implemented. The sizing and synthesis process is an extension of traditional fixed-wing aircraft sizing and synthesis tool. The implemented process can consider vertical flight capacity and provide a take-off weight optimized combination of components of the propulsion system. The main intent of implementing the sizing and synthesis process is to make the framework for the concurrent UAV design and routing problem. Thus, the framework can be extended by integrating other sizing and synthesis tools if the interface is matched.

Lastly, to integrate the UAV design module with the UAV routing module, existing methods have used a sequential approach; after conducting the sizing module, the vehicle routing problem is solved. However, the input and output of the two modules are coupled each other. Thus, the methods cannot address a converged solution for both modules. To alleviate the limitation, this thesis presents a novel modular framework for the concurrent aircraft design and routing problem for sUAS-based delivery systems, which is based on a Fixed Point Iteration (FPI) method to find a converged solution of the coupled problem. The presented framework can provide an optimal vehicle design and routing for the sUAS-based delivery system concurrently.

This thesis uses the developed framework for concurrent UAV design and routing to study a possible package delivery using sUAS in San Diego, CA. The result shows that the developed framework can take into account both planning vehicle operation on the flight

network in which it will be operating and designing the flight network capable of addressing the obstructed environment as part of the vehicle design process.

CHAPTER 1

INTRODUCTION

1.1 Unmanned Aircraft Systems (UAS)

With the emergence of new technologies for small Unmanned Aircraft Systems (sUAS), such as lightweight sensors and high-efficiency batteries, the operation of sUAVs has expanded from military use to commercial use. The UAS market is expected to grow continuously as shown in Figure 1.1. Although the market has been dominated by the demand from public sector (DoD included), commercial demand also has steadily increased. The majority of commercial UAS will fall into the micro and small UAS which will be low-cost and dedicated to specific new and emerging tactical market applications [1]. The potential of utilizing sUAVs has been shown in a wide variety of areas such as military, industry, and even hobbyist fields [2]. In the civilian realm, various applications have been associated with UAS such as environmental research, search and rescue, facility/infrastructure inspection, agriculture, construction survey, payload delivery, and communication relay.

Unmanned Aircraft Systems (UAS) are systems consisting of vehicles, ground stations, and connections among vehicles and ground stations. An Unmanned Aerial Vehicle (UAV) or an Unmanned Aircraft (UA) indicates a vehicle as a component of UAS [3]. More formal definitions can be found in the Department of Defense (DoD) dictionary [4]. According to the DoD dictionary, an unmanned aircraft system is defined as “a system whose components include the necessary equipment, network, and personnel to control an unmanned aircraft” while an unmanned aircraft is defined as “an aircraft that does not carry a human operator and is capable of flight with or without human remote control.” According to [5], a UAV is specifically defined as “A powered vehicle that does not carry a human operator, can be operated autonomously or remotely, can be expendable or recoverable, and can carry a

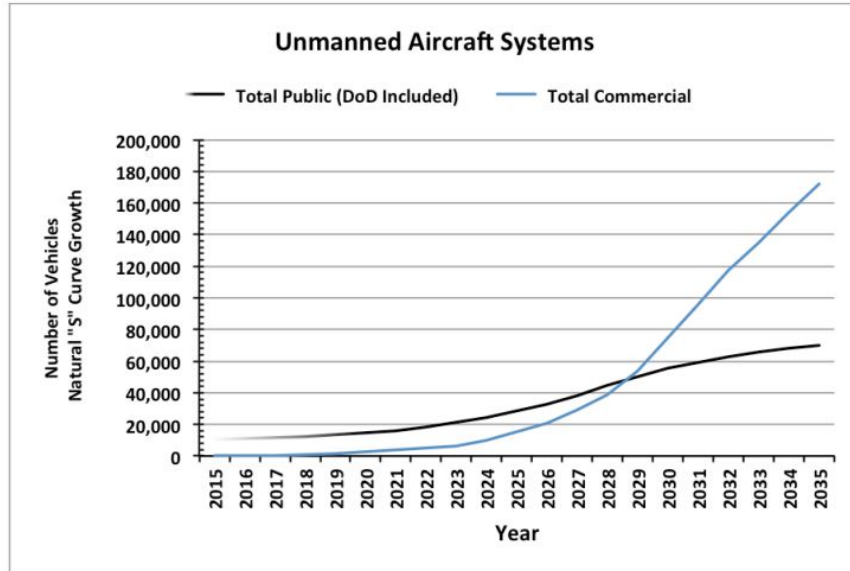


Figure 1.1: Total UAS forecast 2015 - 2035 [1]

lethal or nonlethal payload. Ballistic or semi-ballistic vehicle, cruise missiles, artillery projectiles, torpedoes, mines, satellites, and unattended sensors (with no form of propulsion) are not considered unmanned vehicles. Unmanned vehicles are the primary component of unmanned systems.”

Unmanned Aircraft Systems (UAS) platforms can be categorized by flight altitude and flight endurance such as Low-Altitude Short-Endurance (LASE) or High-Altitude Long-Endurance (HALE) [6]. Figure 1.2 shows typical UAS platforms with flight endurance and flight altitude. Another common classification of UAS platforms is based on vehicle’s weight like nano UAS or small UAS. Table 1.1 describes each category with typical properties of the platform. Weight is a critical factor to UAS-based package delivery systems because of the way the Federal Aviation Administration (FAA) regulations are written. In general, small UAS conduct LASE missions while UAS with larger aircraft such as Global Hawk execute HALE missions. Furthermore, UAVs are grouped by the method on how to generate lift; fixed-wing UAVs, rotary UAVs, or hybrid UAVs which have fixed wings with the Vertical Take-Off and Landing (VTOL) capability.

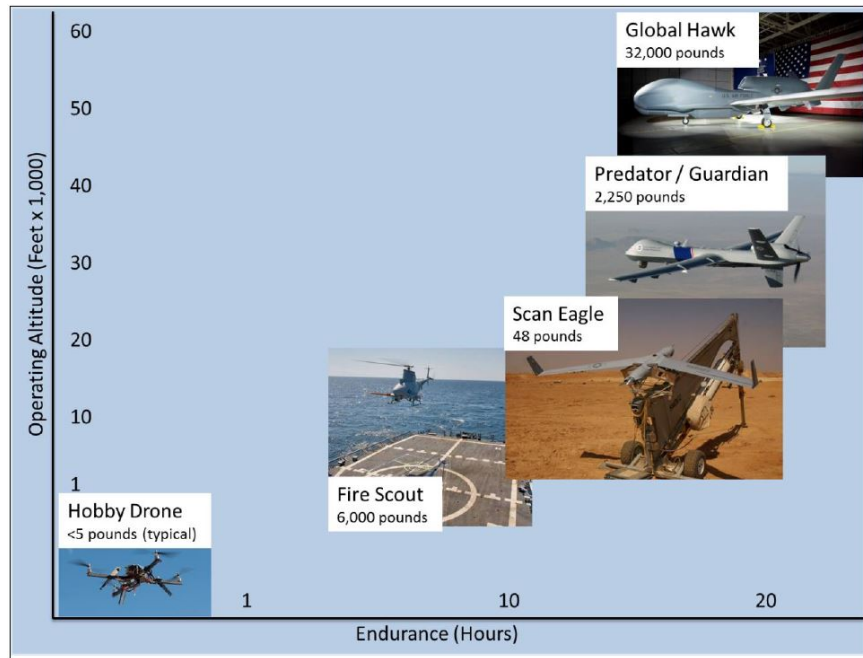


Figure 1.2: Operating altitudes, endurance, and weight of common UAS [7]

Table 1.1: Notional UAS vehicle classification and categorization [1]

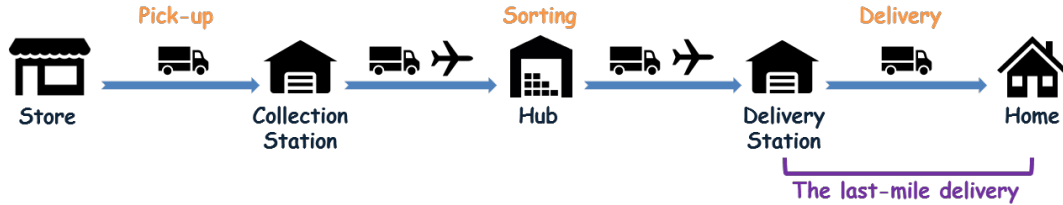
UAS Description	Weight (Pounds)	Overall Size (Feet)	Mission Altitude (Feet Above the Surface)	Mission Speed (Miles per Hour)	Mission Radius (Miles)	Mission Endurance (Hours)
Nano	< 1	< 1	< 400	< 25	< 1	< 1
Micro	1 to 4.5	< 3	< 3,000	10 to 25	1 to 5	1
Small UAS	4.5 to 55	< 10	< 10,000	50 to 75	5 to 25	1 to 4
Ultralight Aircraft*	55 to 255	< 30	< 15,000	75 to 150	25 to 75	4 to 6
Light Sport Aircraft*	255 to 1320	< 45	< 18,000	75 to 150	50 to 100	6 to 12
Small Aircraft*	1,320 to 12,500	< 60	< 25,000	100 to 200	100 to 200	24 to 36
Medium Aircraft*	12,500 to 41,000	TBD	< 100,000	TBD	TBD	TBD

* FAA-defined manned aircraft weight categories

□ Order processing



□ Delivery processing



□ The last-mile delivery

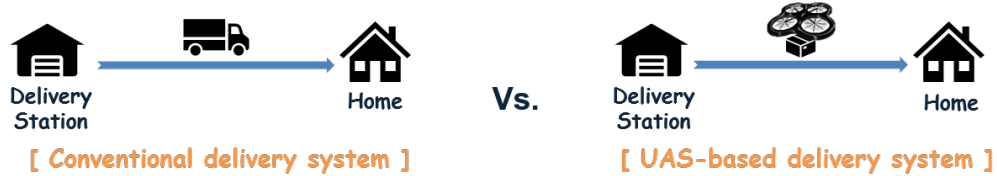


Figure 1.3: Notional delivery system

1.2 Delivery Systems based on a sUAS

In recent years, the last-mile package delivery problem has emerged as one of the main industry applications for sUAS [8, 9, 10, 11]. A notional delivery system is illustrated in Figure 1.3. If a customer orders items on an online shopping site, a delivery company picks up the items and moves them to a hub to sort them. After sorting items, the company moves them to a delivery station. Lastly, the items are moved to the customer, and this step is called the last-mile package delivery. In the conventional delivery system, the last-mile package delivery is conducted by vans and trucks. Recently, sUAS has considered as an alternative for the last-mile package delivery. First, a sUAV has the considerable potential for reducing the acquisition and operating costs for package deliveries in comparison to vans or trucks [11]. Second, a sUAV is capable of delivering packages faster: its travel distance is significantly shorter than surface streets due to almost direct flight route. Moreover, the UAS-based operation is more predictable because of not relying on the complex traffic sys-



Figure 1.4: The developing sVTOLs for a package delivery mission [12, 13, 14, 15, 16, 17]

tem. Because of these benefits, sUAS has the potential of satisfying the emerging demand for building better last-mile delivery systems from package delivery companies such as Amazon, UPS, DHL, and FedEx. For the package delivery missions, sVTOL UAVs have been considered by various companies as shown in Figure 1.4.

1.3 Aircraft Sizing and Synthesis Process

To address perspectives on aircraft design for delivery missions, this section introduces both a traditional aircraft sizing and synthesis process for a single-vehicle mission and an aircraft sizing and synthesis process for a fleet-level mission. A traditional aircraft sizing and synthesis process mainly deals with a relatively simple Concept of Operations (ConOps) while an aircraft sizing and synthesis process for a fleet-level mission considers the complex concept of operations which cannot be determined intuitively [18].

1.3.1 Traditional Aircraft Sizing and Synthesis Process

The traditional aircraft sizing and synthesis process has focused on how to obtain the design parameters of an aircraft which is able to complete the given missions. Raymer [19] and Mattingly [20] provide the traditional sizing and synthesis process for a fixed-wing aircraft. For instance, let us consider an aircraft sizing problem for a delivery mission including four mission points as illustrated in Figure 1.5 and assume that we need to find the route which has the shortest total distance moved by a vehicle to visit each mission point once. The ConOps of the given mission can be determined intuitively because the mission is relative simple, and a mission profile can be built based on the ConOps. Then, the traditional aircraft sizing and synthesis process is executed with aerodynamic, propulsion, and structure models. The traditional process consists of two primary analyses: constraint analysis and weight analysis/estimation. Constraint analysis finds feasible design space using aircraft performance constraints which are described by thrust loading T/W , or power loading P/W , and wing loading W/S . Weight analysis estimates an optimal take-off weight which allows to complete a given mission based on a mission profile. These two processes are linked by a iteration method called a fixed-point iteration [21].

The traditional aircraft sizing and synthesis process is designed for fueled fixed-wing aircraft. However, according to [22], as weight and cost of electrical batteries have decreased, the development of electric-powered small UAVs has substantially increased. In this sense, the traditional sizing process has expanded to include various concepts of the propulsion system. To address a sizing and synthesis process for revolutionary concept aircraft, Nam [23], as seen in Figure 1.6, extends Raymer and Mattingly's process [19, 20] that address fueled fixed-wing aircraft only. By modeling the propulsion system and fuel as an integration of power paths and the source of energy respectively, the process is capable of sizing an aircraft having solar cells or electric batteries as well as traditional fueled aircrafts. Gundlach [24] also presents a sizing process as shown in Figure 1.7, which is an extension of Raymer and Mattingly's process with electric aircraft performance equations

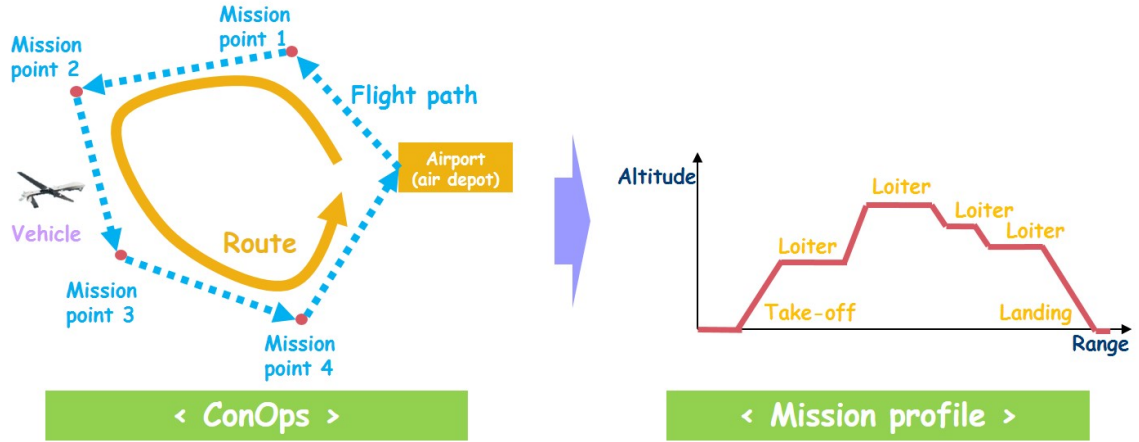


Figure 1.5: The Concept of Operations (ConOps) and mission profile of a delivery mission

for electric-powered aircraft which describe the relation between battery mass fraction and endurance/range. However, the limitation of these extended sizing and synthesis processes of Nam [23] and Gundlach [24] is that they work with fixed-wing aircraft only while UAS-based delivery systems mainly utilize sVTOL UAVs. Thus, to address the UAS-based delivery systems, this thesis presents a sizing and synthesis process for sVTOL UAVs.

These processes [19, 20, 23, 24] have addressed the relation between subsystems (or components) and a system (or aircraft) to obtain the optimal design parameters to complete a given mission such as thrust loading T/W , power loading P/W , wing loading W/S , and take-off weight. They have mainly focused on a single vehicle mission that can determine the ConOps intuitively. However, a fleet-level mission is hard to intuitively determine the ConOps due to its complexity. The next subsection discuss more details about the sizing and synthesis process for a fleet-level mission.

1.3.2 Aircraft Sizing and Synthesis Process for a fleet-level mission

To address a fleet-level mission, let us consider the delivery mission in previous section again. However, in this time, there exist more mission points to be visited, enough so that the optimal paths are not able to be selected simply, and multiple homogeneous vehicles for the mission as shown in Figure 1.8. In order to conduct the sizing and synthesis process,

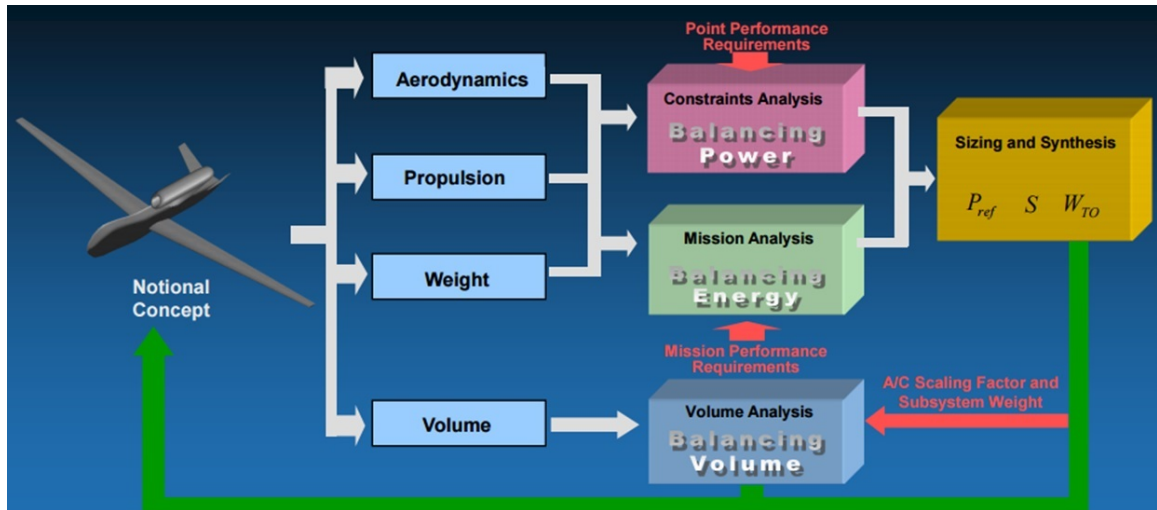


Figure 1.6: Nam's sizing method [23]

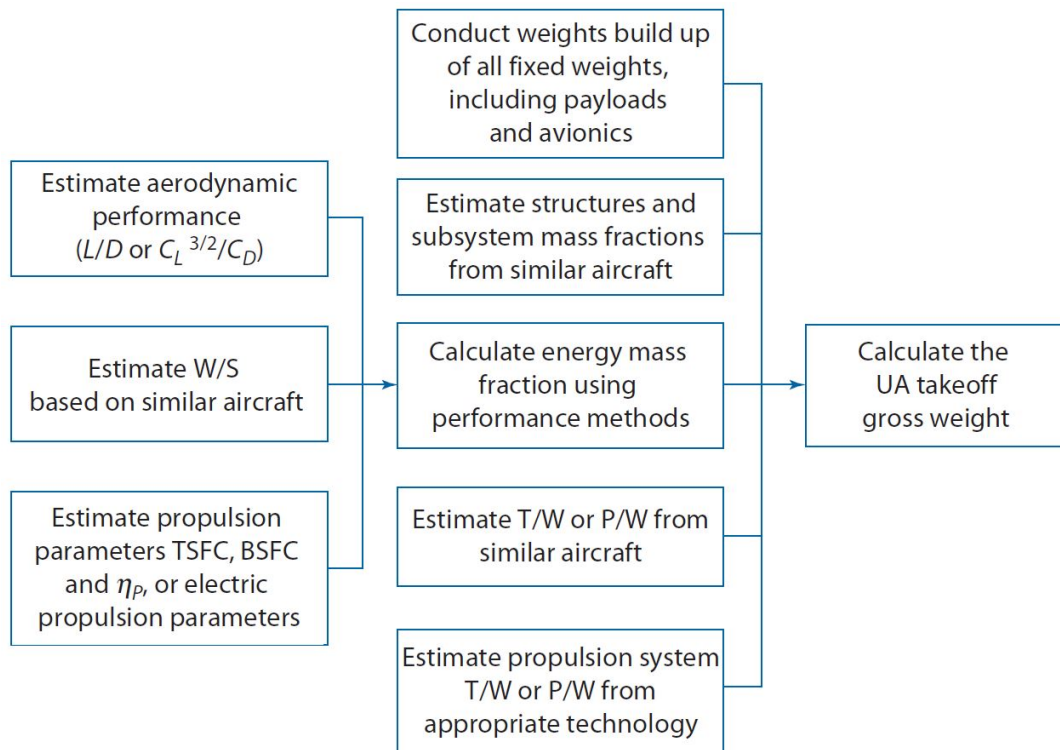


Figure 1.7: Gundlach's sizing method [24]

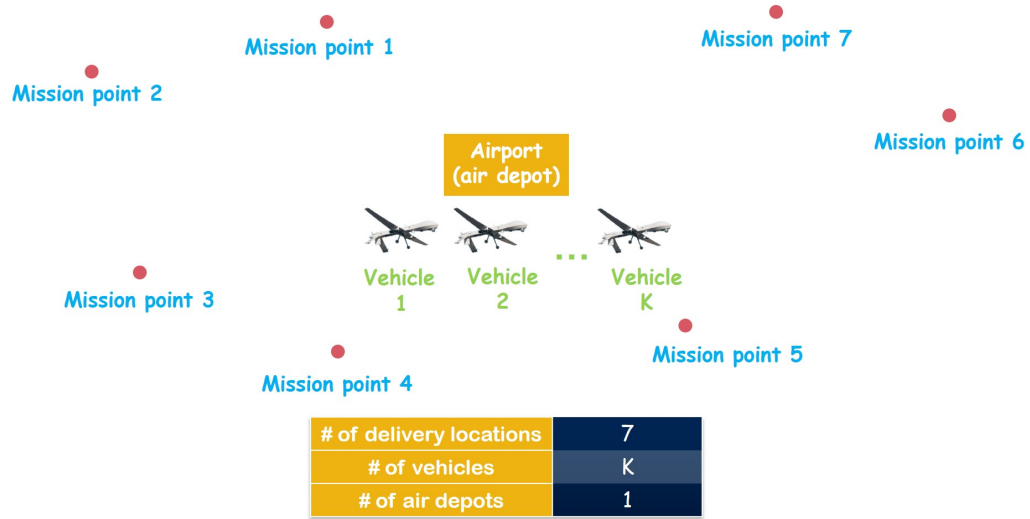


Figure 1.8: Example of a delivery mission requiring multiple UAVs

the set of routes, which has the shortest total distance moved by vehicles and visits each mission point once, needs to be found. Then, mission profiles are created which describe the ConOps for each vehicle to complete a mission. For a single UAV mission, an optimal ConOps could be simply determined. However, for a multi-UAV mission, a set of the optimal ConOps for each UAV is hard to be determined due to its complexity; there are at least two questions should be answered: how many UAVs are required, and which mission points are visited by a UAV? Figure 1.9 and 1.10 illustrate two possible ConOps for the multi-UAV delivery mission including seven mission points. If position values of each point are known, the better ConOps can be found in terms of total distance moved by vehicles. However, the number of possible scenarios increases exponentially as the number of either mission points or UAVs increases.

Determining an optimal ConOps for a fleet-level mission can be executed by solving an optimization problem called the vehicle routing problem (VRP) which has been actively studied in Operations Research (OR) field [25]. The VRP is a generalized variant of the traveling salesman problem (TSP). A TSP can determine the optimal order of cities visited by a salesman. For a given fleet-level mission, a VRP can determine the number of required vehicles and a set of routes for each vehicle based on its objective function such as the

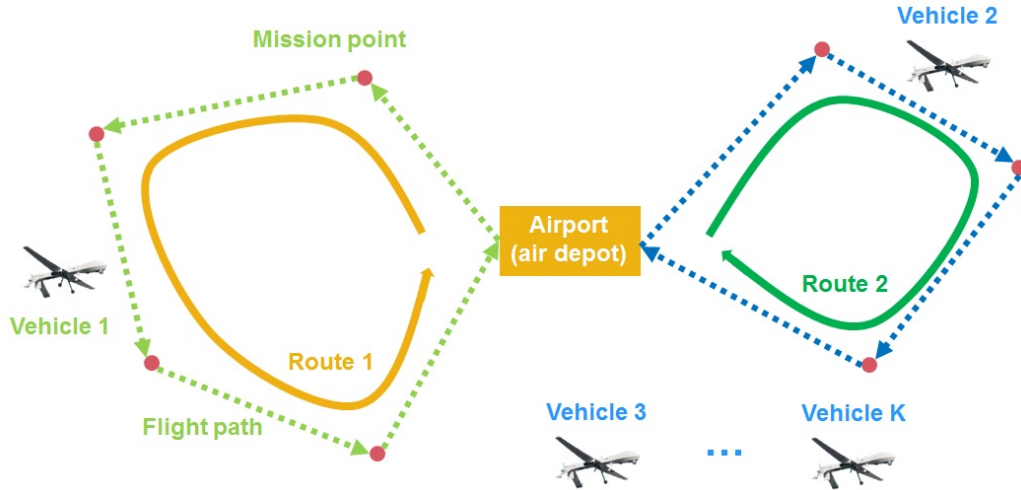


Figure 1.9: The first possible optimal ConOps for a fleet-level delivery mission including seven mission points

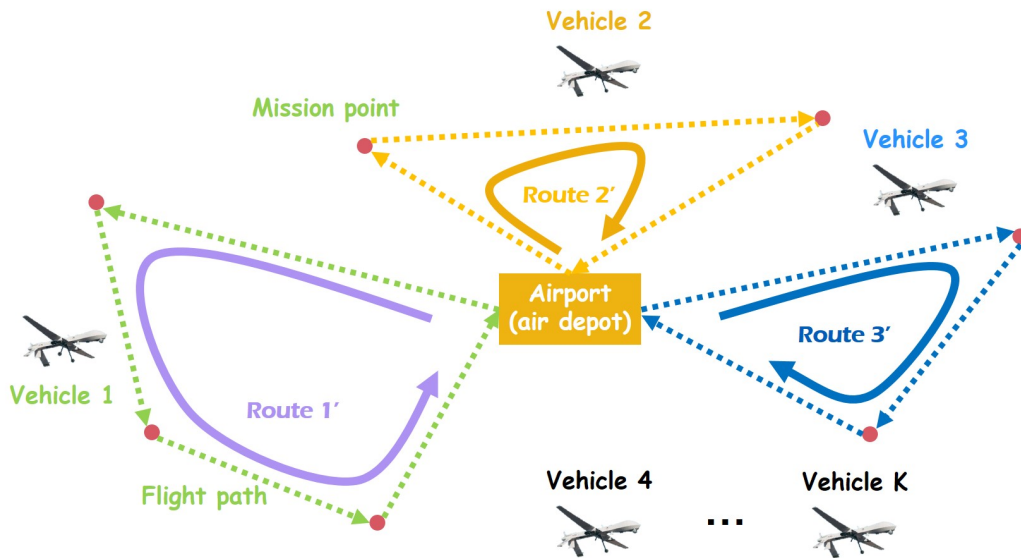


Figure 1.10: The second possible optimal ConOps for a fleet-level delivery mission including seven mission points

minimum of total distance or flight time tripped by vehicles and constraints such as the maximum of endurance and range of vehicles. After creating vehicle routes describing each ConOps, mission profiles can be created based on each route. Thus, the VRP can be utilized to obtain a set of the optimal ConOps systematically. Based on each ConOps, a mission profile can be created as illustrated in Figure 1.11. Then, the mission profiles can be utilized for a traditional sizing and synthesis process to obtain the optimal design

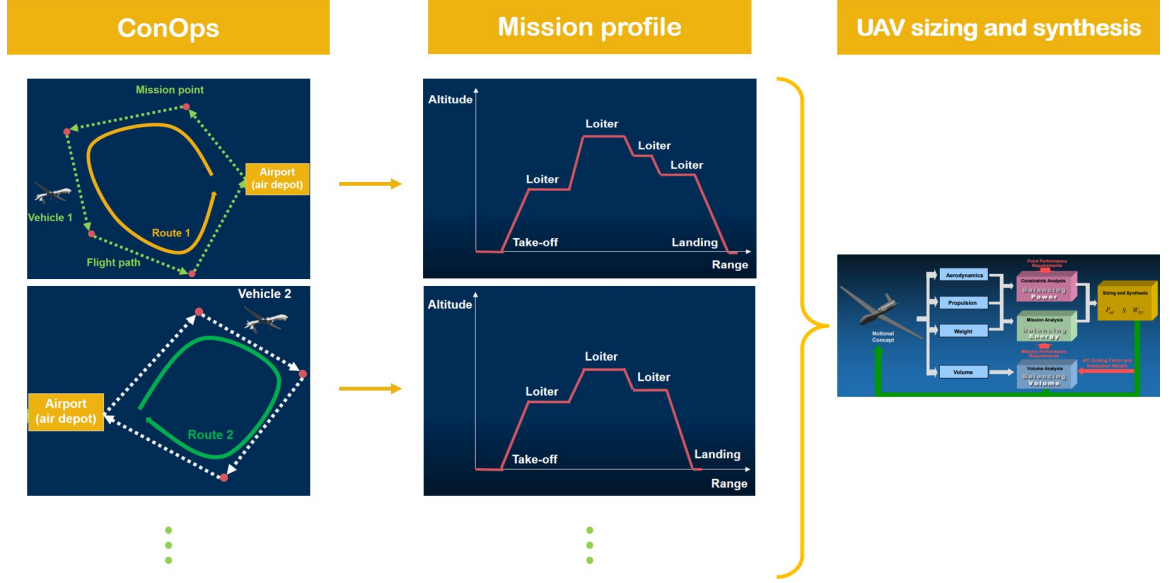


Figure 1.11: Notional sizing and synthesis process with a fleet-level mission (UAV sizing and synthesis block is adopted from [23])

parameters of a vehicle.

Therefore, a fleet-level sizing and synthesis process for a delivery mission can be described by a combination of two optimization problems: a vehicle routing problem and a sizing and synthesis process. A vehicle routing problem optimizes the operations of a UAS such as the number of required vehicles and routes for each vehicle while a sizing process explores vehicle design space to find optimal design parameters of a UAV such as wing loading, thrust/power loading, and take-off weight. That is, these two optimization problems are utilized to minimize operating cost and acquisition cost of a UAS as illustrated in Figure 1.12. In this thesis, the total cost is defined as

$$\text{Total cost} = \frac{\text{acquisition cost}}{\# \text{ of days in operational life}} + \text{costs for one day of operations} \quad (1.1)$$

This definition can be used for a VRP as an objective function directly.

From the fleet-level delivery mission example, a sizing and synthesis process of a UAV for a UAS-based delivery system can be described by a combination of a UAV sizing and synthesis process and a VRP. Therefore, this thesis focuses on how to concurrently solve a

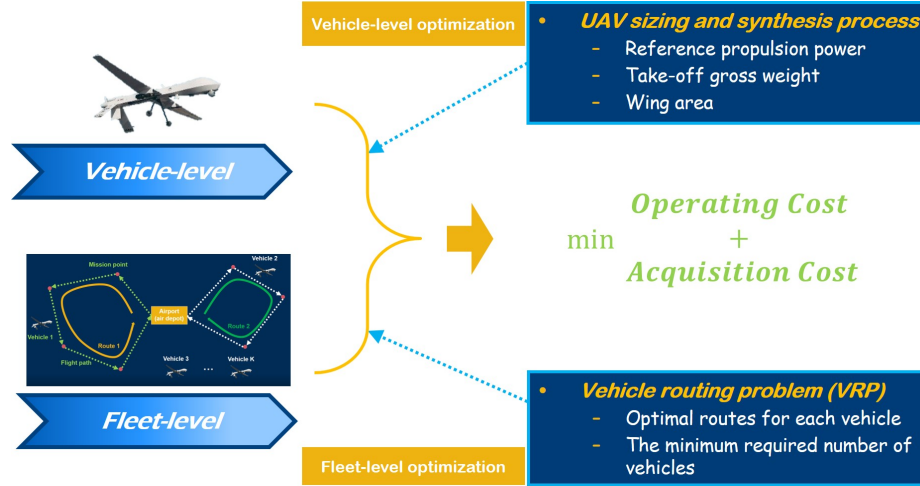


Figure 1.12: Two approaches to optimizing a plan for a fleet-level mission

sizing and synthesis process and a VRP to obtain total cost.

1.4 Challenges for the operation of sUAS-based delivery systems

A key challenge for the operation of a sUAS-based package delivery system in an urban area is creating viable operating plans. It should satisfy constraints from vehicle's properties and logistics characteristics as well as provide feasible schedules for vehicles in mission areas. In general, the planning for delivery systems can be modeled as a vehicle routing problem (VRP). The result of the VRP leads total cost optimized schedules for vehicles based on an objective/cost function and constraints. The VRP and its derivatives allow the specific modeling of vehicle properties, logistics concepts, and flight path planning in order to capture appropriate physical concepts, business rules, and regulatory requirements under Part 107 [26] which is the Federal Aviation Administration (FAA) regulations for small UAS operations.

Small UAVs have intrinsic operational limitations which must be appropriately modeled in the VRP [27]. While flight times of sUAVs are much shorter than the driving time of traditional transport vehicles, they generally carry very few packages. These characteristics drive a sUAS-based delivery system to repeatedly operate sUAVs during operating

hours. Additionally, sUAS-based delivery systems will be asked to provide delivery services to customers in an on-demand fashion during specific scheduled time windows [9]; this is a new type of service that can be enabled by sUAVs because of the better temporal predictability of an air delivery network compared to existing urban ground delivery systems. Finally, the most common power resource of sUAVs is rechargeable batteries [28]; This will require a strategy for recharging or replacing batteries in order to maximize the rate of operation of sUAVs during a delivery operation.

1.5 Research Motivation

The aforementioned fleet-level delivery problem is called a concurrent vehicle design and routing problem including two optimization processes: a vehicle routing problem and a sizing and synthesis process [29, 30, 31, 32, 33]. According to Mane et al. [29], a concurrent vehicle design and routing problem is a Mixed-Integer Non-Linear Programming (MINLP) problem; a concurrent problem consists of a vehicle routing problem, which is a Mixed-Integer Linear Programming (MILP) problem having integer and real/continuous design values, and a UAV design/sizing problem, which is a nonlinear process including nonlinear equations.

In order to solve an MINLP problem, decomposition methods have been successfully utilized [29, 30, 31, 32, 33]. Taylor and Weck [30] present an Multi-Disciplinary Optimization (MDO) model to address a concurrent vehicle design and routing problem for a fleet-level mission consisting of both the transportation network flow and the vehicle design problems. The MDO model consists of the system-level optimizer, the vehicle design module, the network flow and operations modules as shown in Figure 1.13. The system-level optimizer is implemented to use a simulated annealing method which is a metaheuristic method as illustrated in Figure 1.14. In order to reduce the problem size, they assume that aircraft can fly only round trips between two cities while packages can travel through other cities toward the destination. The vehicle design module is implemented by numerical

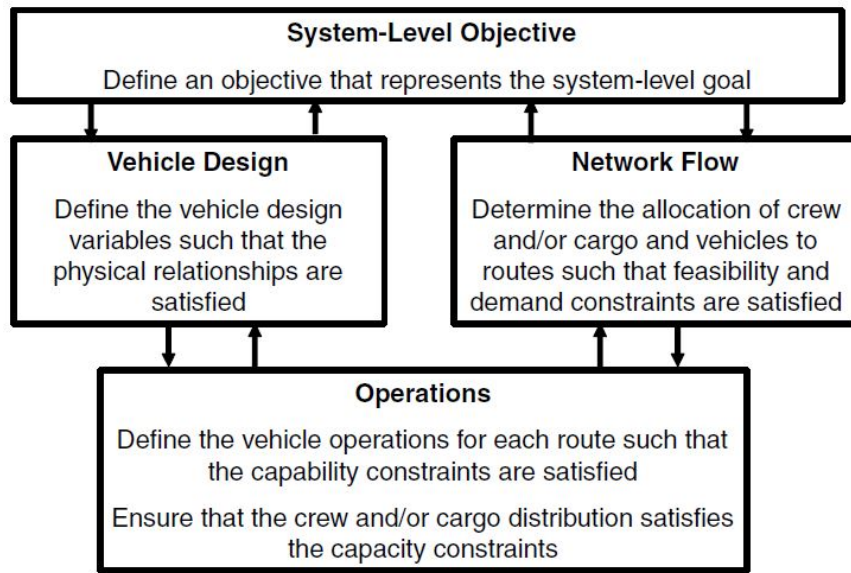


Figure 1.13: Diagram of the integrated transportation system model [30]

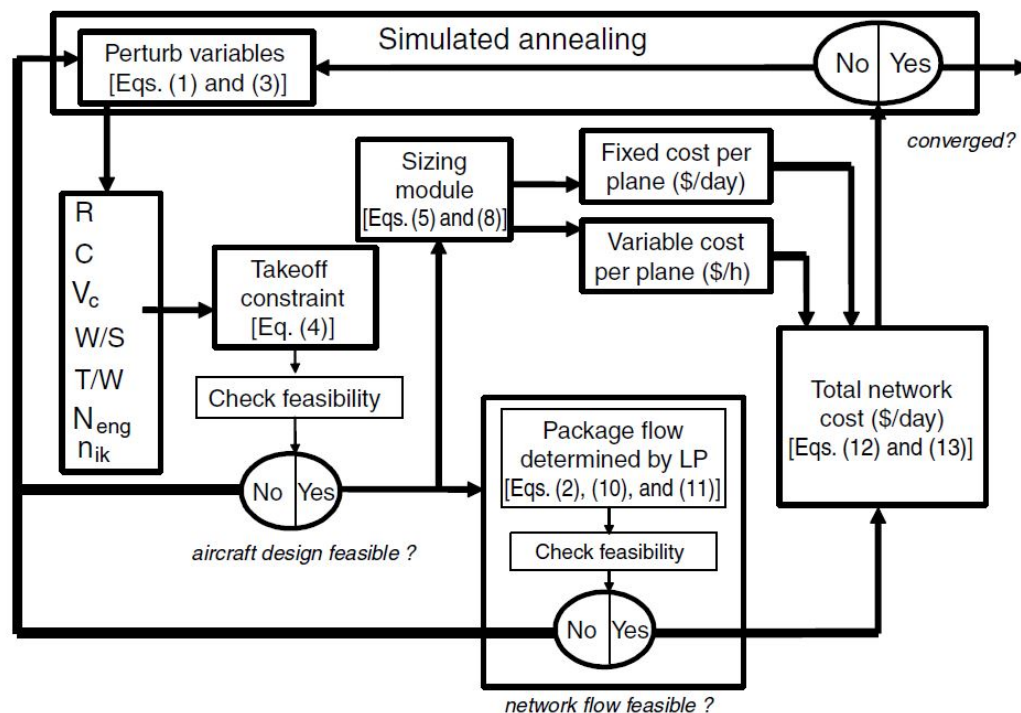


Figure 1.14: Integrated transportation system design optimization with simulated annealing [30]

computations.

Mane et al. [29] present a bi-level MDO model consisting of the system-level optimizer and two subsystem-level optimizers for the aircraft sizing and the airline allocation subproblems as shown in Figure 1.15. After the cost coefficients for the new vehicle are obtained by solving an aircraft sizing subproblem, the values are utilized to solve an airline allocation subproblem. Then, the system-level optimizer calculates the minimum operating cost based on the results from subsystems. Later, the MDO model is extended by [31], which can solve a trip assignment problem with an uncertainty model of trip demands instead of an airline allocation problem as shown in Figure 1.16. To address the uncertainty, the Monte Carlo simulation is utilized. Govindaraju and Crossley [32] present an MDO model which extends the MDO model of [31]. In the MDO model, both uncertainty models in operations and aircraft design parameters are integrated as described in Figure 1.17. The Monte Carlo simulation is used to deal with the uncertainty of the aircraft sizing subproblem while the descriptive sampling is utilized to address the uncertainty of the trip assignment subproblem. The descriptive sampling deterministically selects samples using input sample values and their random permutations to reduce the variance of simulation estimates based on the known input distribution [34, 35].

Bociaga and Crossley [33] present a MDO model consisting of three modules: fleet allocation, aircraft sizing, simulation modules for a Coverage Path Planning (CPP) problem, which finds an optimal path covering all points in an Area Of Interest (AOI) as illustrated in Figure 1.18. The fleet allocation is conducted by the saturated D-optimal Design of Experiments (DoE). After executing aircraft sizing, a flight simulation by the designed aircraft is run to calculate mission time and cost.

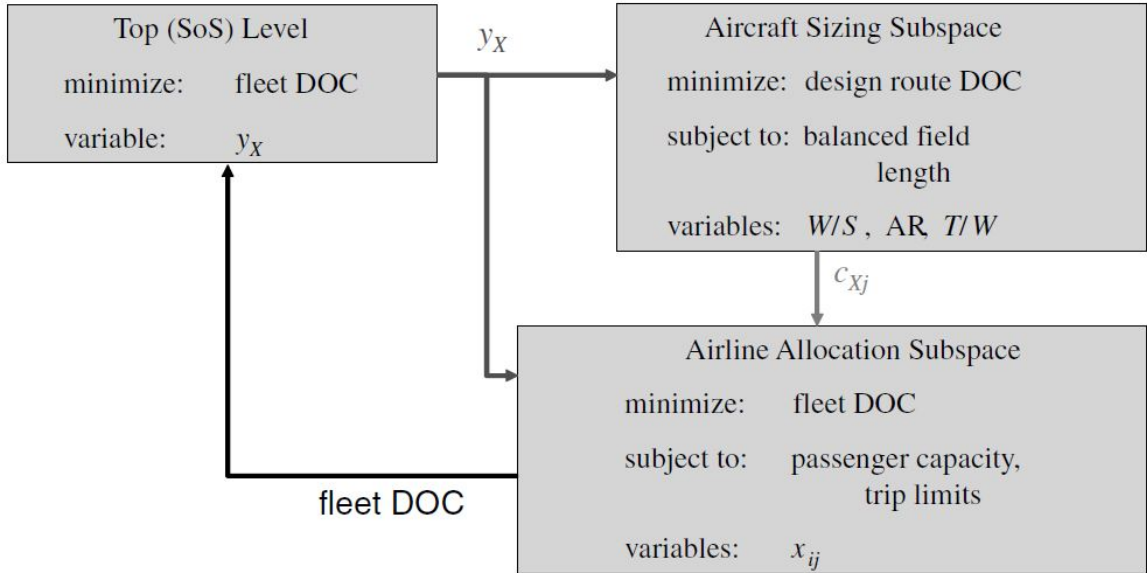


Figure 1.15: Sequential decomposition schematic [29]

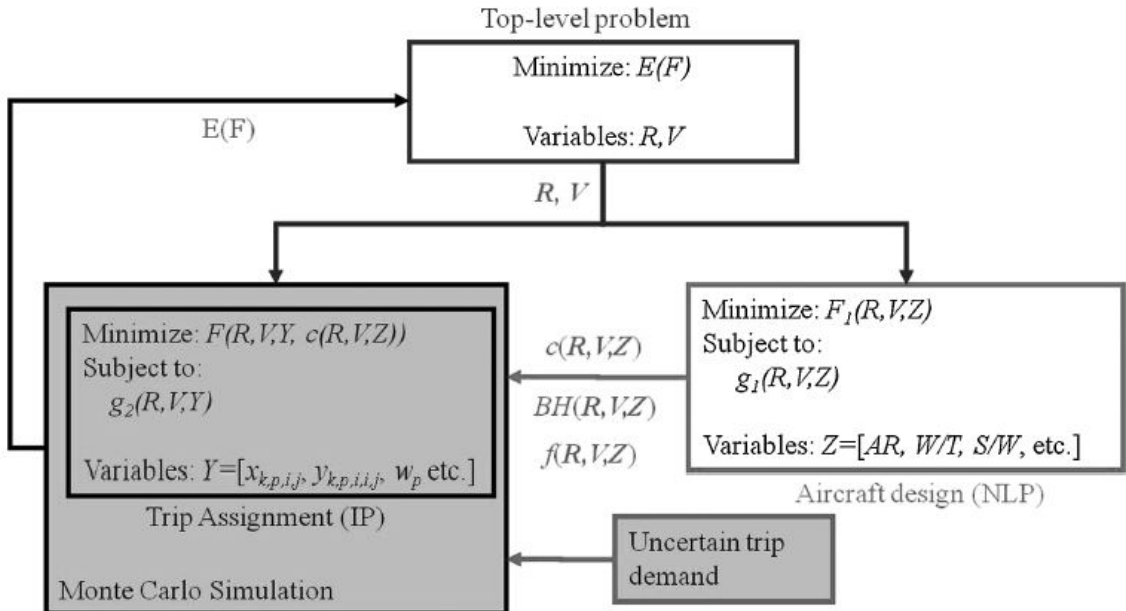


Figure 1.16: Problem formulation and decomposition with uncertainty [29, 31]

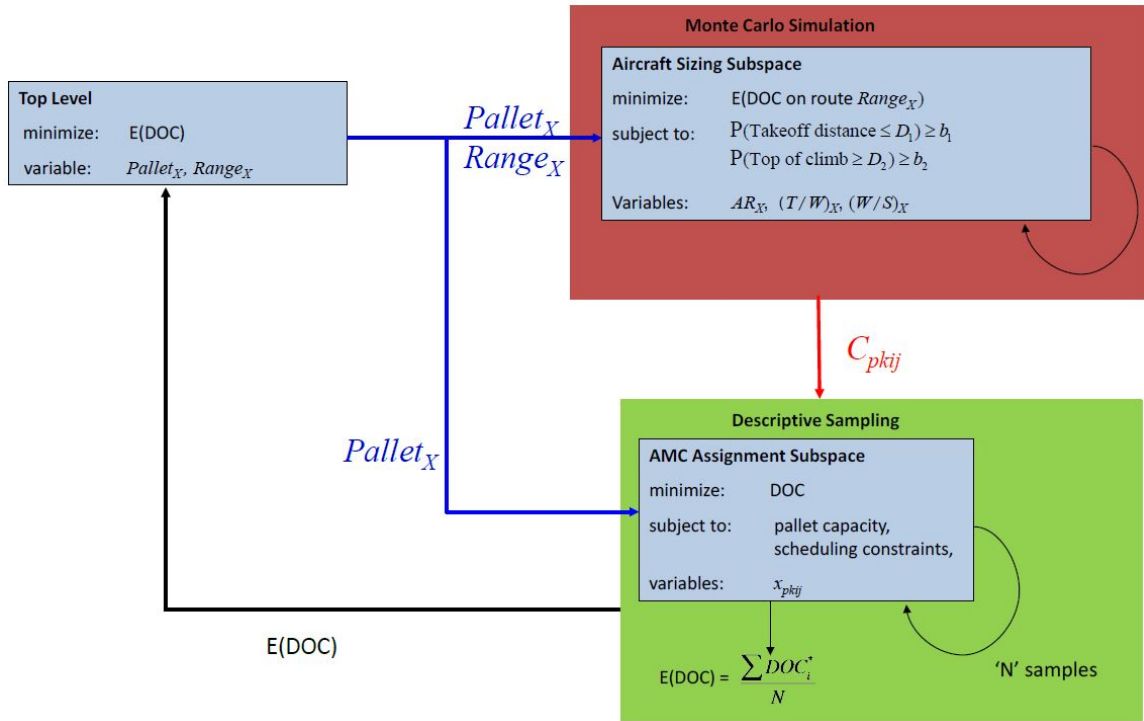


Figure 1.17: Decomposition framework [32]

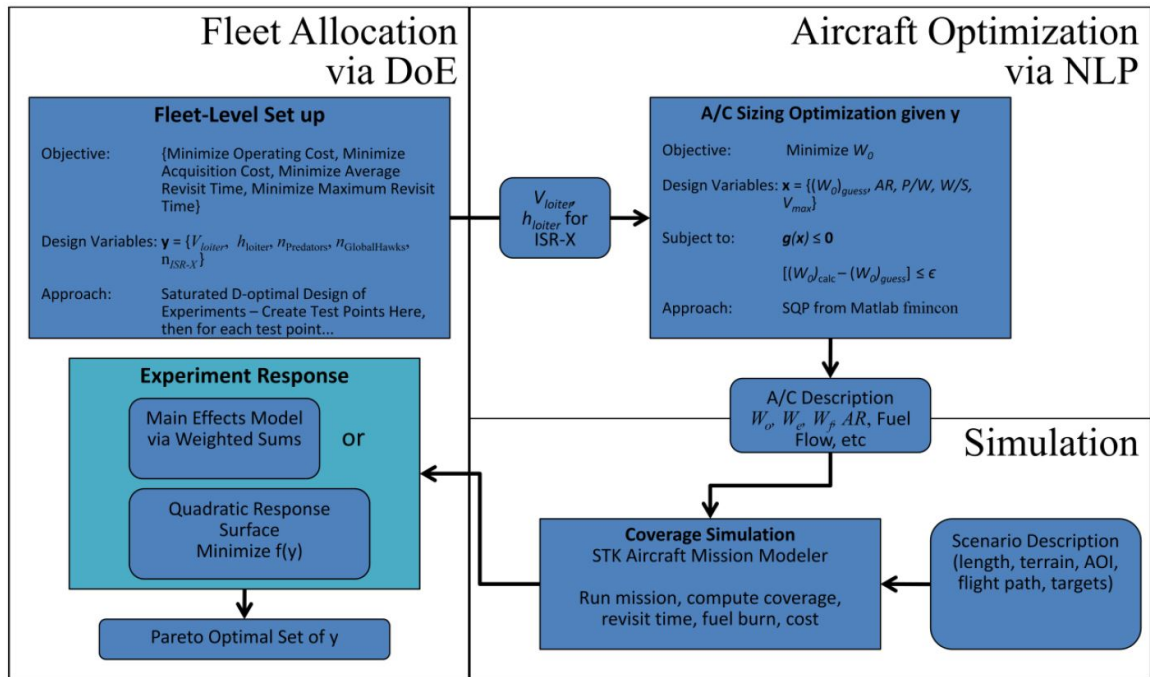


Figure 1.18: Simultaneous Aircraft Sizing and Allocation Decomposition [33]

Table 1.2: Summary of literatures for the concurrent aircraft design and vehicle operation problem

Author	Vehicle type	Network type	Origin-dest. pair	Advantage	Disadvantage
Taylor [30]	Fueled a/c	Air cargo	Airport-to-airport	<ul style="list-style-type: none"> • Be fast in computation time • Provides a set of routes 	<ul style="list-style-type: none"> • Provides an approximated solution • Used symmetric demand • Considered only take-off constraint
Mane [31]	Fueled a/c	Airline	Airport-to-airport	<ul style="list-style-type: none"> • Provides an MDO architecture • Be able to consider an uncertainty of demand 	<ul style="list-style-type: none"> • Be available for traditional fueled aircrafts
Govindaraju [32]	Fueled a/c	Air cargo	Airport-to-airport	<ul style="list-style-type: none"> • Be able to consider both uncertainty in operations and design parameters 	<ul style="list-style-type: none"> • Considered only fleet assignment problem, not VRP
Bociaga [33]	UAV	Area coverage	Airport-to-airport	<ul style="list-style-type: none"> • Provides a new architecture with simulation steps • Combined a new type of missions 	<ul style="list-style-type: none"> • Considered fleet assignment problem • Be available for only fueled UAVs

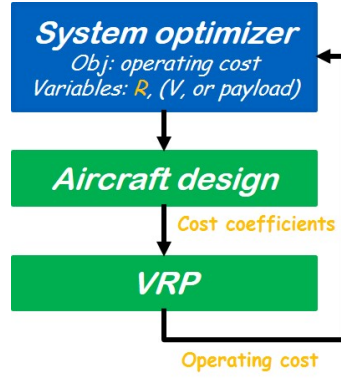


Figure 1.19: Simplified structure of the MDO models [29, 30, 31, 32, 33]

The aforementioned MDO models [30, 31, 32, 33] are summarized in Table 1.2. The first observation is that the structures of the MDO models can be simplified as illustrated in Figure 1.19. The procedure can be described as follows

- Step 1: Range/speed is fixed by system optimizer
- Step 2: Aircraft sizing is conducted with a given range/speed
- Step 3: Based on specifications of the designed aircraft, cost coefficients of a VRP are updated
- Step 4: By solving the VRP, operating cost/objective function is calculated
- Step 5: System optimizer checks optimality of the design variables

The observation from this structure is that this decomposition approach consists of an aircraft design module and a VRP module, and the modules are linked sequentially. The limitation of this structure is that the VRP module cannot have direct effects on the aircraft design module because there is no direct feedback loop.

To address the limitation, the input and output of each module described in Table 1.3 need to be observed. The UAV design module requires a mission profile and assumption models for the sizing and synthesis process. After the aircraft design parameters are feasible and converged, the UAV design module provides a set of optimal design values which determine the vehicle specification.

Table 1.3: Input and output of the UAV design module and vehicle routing module

Module	Input	Output
UAV design	<ul style="list-style-type: none"> • Mission profile • Assumption models <ul style="list-style-type: none"> ▷ Aerodynamics ▷ Propulsion ▷ Weight 	<ul style="list-style-type: none"> • Vehicle specification
Vehicle routing	<ul style="list-style-type: none"> • Vehicle specification • Set of customers and a depot • Set of demands • Flight network 	<ul style="list-style-type: none"> • Set of routes • The required number of vehicles



Figure 1.20: Interaction between the UAV design and vehicle routing modules

On the other hand, the vehicle routing module requires sets of customers, depots, and demands in order to create feasible routes in a logistical point of view. A flight network is also an important input of the vehicle routing module because all the candidates of flight paths created only based on the flight network; this is shown in Chapter 3 and 4. Using this input data, the vehicle routing module creates a set of total cost optimized routes with the number of required vehicles to conduct the operations plan.

There, however, is an important characteristic between the inputs and outputs of the UAV design and vehicle routing modules. A route from the vehicle routing module shows a flight schedule of a vehicle, which is able to be converted into a mission profile. That is, after a set of total cost optimized routes are built by the vehicle routing module, a mission profile can be created by one of the routes. The vehicle parameters required for the vehicle routing module also can be determined by the UAV design module. Thus, the two individual modules are able to be transformed into a coupled process as shown in Figure 1.20. Thus, the feedback loop from the vehicle routing module to the UAV design module

should be considered to solve a concurrent vehicle design and routing problem.

The second observation from the MDO models [30, 31, 32, 33] is that their aircraft design modules mainly consider fixed-wing aircraft sizing and synthesis. However, various companies have developed UAS-based delivery systems utilizing sVTOL UAVs such as Amazon, DHL, UPS, and Google. For urban delivery, vertical take-off capacity not requiring a runway is essential. To alleviate this limitation of the MDO models, the aircraft design modules should be extended to address design parameters of a sVTOL UAV such as power loading, wing/disk loading, and take-off weight.

The third observation is that the vehicle routing modules of the models address simple VRP formulations not including both a time notion and reuse of vehicles. However, these notions are substantial factors when formulating operations of UAVs. Thus, the vehicle routing modules should involve time and reuse notions to deal with a UAS-based delivery mission.

The last observation is that the MDO models consider point-to-point/airport-to-airport flight networks based on the Euclidean distance. However, the flight networks cannot describe the obstructed environment such as an urban area because the flight networks do not include any intermediate information of each path. For a UAS-based urban delivery, a novel flight network which can address collision-free paths in an urban area needs to be.

These observations directly motivate this thesis to research and develop a framework for concurrent UAV design and routing for urban delivery systems illustrated in Figure 1.21. To alleviate the limitations of aforementioned MDO models [30, 31, 32, 33], this framework extends their decomposition approaches in terms of

- Allowing to design a small VTOL UAV in a UAV design module
- Formulating a VRP for UAS-based delivery systems
- Including an urban model to address collision-free paths
- Finding a solution of a coupled problem of the small VTOL UAV design module and the VRP module for urban delivery

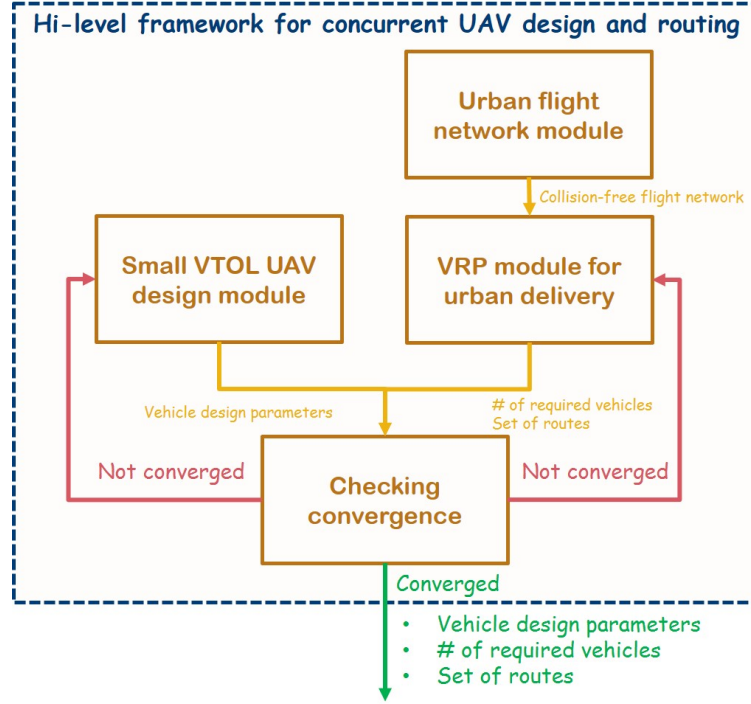


Figure 1.21: Hi-level framework for concurrent UAV design and routing for urban delivery systems

Each extension is associated with a research question of this thesis that will be presented in the end of this chapter.

In addition, because the decomposition approach deals with each module as a black box, if the interface among modules is matched, each module in Figure 1.21 can be implemented by any method or algorithm. This is an advantage of a framework based on a decomposition approach.

1.6 Research Scope

To address a more practical problem, this section narrows down research scope of the extensions of the framework for concurrent UAV design and routing for urban delivery systems presented in the previous section.

First of all, the vehicle type considered in this thesis is a small fixed-wing vertical take-off and landing (VTOL) UAV having relatively long range and vertical flight capacity not

requiring a runway. The vertical flight capacity is necessary to delivery a package safely. Specifically, a tiltrotor of DHL using the wing-copter platform is used a baseline design for a vehicle design process. However, to address an aircraft sizing process based on off-the-shelf components, a tiltrotor which uses 8 rotors for vertical flight and 4 rotors for forward flight is considered.

Second, this thesis deals with a VRP for UAS-based delivery systems that finds a total cost optimized solution in terms of the minimum of total cost (Equation 1.1) with constraints such as

- Temporal constraints (i.e. available time periods for delivery)
- Limitations for vehicles' performance in terms of payload capacity and range/endurance
- Reuse of sVTOL UAVs

Third, this thesis considers an urban environment when creating a flight path/network to operate UAVs in an urban area. An urban environment adds new matters which are not considered in conventional air transport network problems into the optimization problem such as

- How to meet regulatory requirements
- How to create feasible/collision-free flight paths
- How to select delivery locations
- How to maintain a link among the vehicles and the ground station

The UAS also should satisfy the regulatory requirements under Part 107 [26] of the Federal Aviation Regulations (FARs) as seen in Table B.1 in Appendix B. For a mission in urban areas, there are the highest two barriers under the regulation such that

- Visual Line-Of-Sight (VLOS) operation is allowed only
- Small unmanned aircraft may not operate over any person not directly participating in the operation, under a covered structure, and inside a covered stationary vehicle

The VLOS operation requires that each UAV should remain within VLOS of a pilot or a visual observer. This requirement increases the operating cost drastically. Thus, this work assumes that Beyond Visual Line-Of-Sight (BVLOS) operation which autonomously operates UAVs without observers is allowed, and that UAVs are allowed to operate over any person in a mission area except for preassigned restricted areas such as airports, large public parks, and military areas. Thus, to address operations of UAVs in an urban environment, collision-free path-planning is required to avoid urban obstacles and restricted airspace. In the real applications, Temporary Flight Restrictions (TFRs) should be considered, but this thesis does not address TFRs because they can be dealt with as restricted areas if needed.

Conventional air network problems have utilized straight paths representing the Euclidean distance from airport A to airport B to estimate the operating cost. Since the Euclidean path cannot address collision-free paths in urban areas, this thesis presents a novel framework to create an urban flight network.

Selecting delivery locations also is a substantial factor to operate a UAS in an urban environment since UAV's access capability should be considered. In this work, delivery companies' offices where can provide a landing spot are used as delivery locations because in an urban area, there are many tall buildings or apartments which cannot directly visited to customers by a sVTOL UAV. Thus, the scenario that customers should visit a delivery companies' office to pick up their packages is considered. This allows to reduce the problem size of a VRP by decreasing the number of mission points.

For an urban mission, maintaining a communication link among the vehicles and the ground station is required. To do that, this thesis assumes that each delivery location has a link with the ground station to share vehicles' information. This assumption allows UAVs to be linked with the ground station in most of mission time because in an urban delivery mission, there are many mission points linked with the ground station in a relatively narrow mission area. Thus, the BVLOS operation also can be allowed by the assumption.

Finally, this thesis considers a Fixed Point Iterations (FPI) method to solve a coupled

problem of the small VTOL UAV design module and the VRP module for urban delivery. To find a solution of a system of nonlinear equations with a multidisciplinary perspective, the fixed point iteration has been widely used [36, 37, 38]. Especially, in the aircraft design field, the fluid-structure analysis has utilized a FPI method to obtain converged values from both fluid analysis and structural analysis [39, 40], and the weight analysis in the aircraft sizing and synthesis process has used a FPI method to find a converged take-off weight [41]. This thesis takes advantage of a FPI method to acquire converged values from both the small VTOL UAV design module and the VRP module.

1.7 Research Question and Objective

The research observations and the research scopes, as stated previously, inspire research questions of this thesis as follows

Research Question 1 (RQ 1) : How can a set of main components of small fixed-wing VTOL UAVs as a result of a sizing and synthesis process be obtained?

Research Question 2 (RQ 2) : What type of a VRP optimization model best represents the operations of sUAS-based delivery systems?

Research Question 3 (RQ 3) : What type of an input graph of a VRP model best represents collision-free paths in the urban environment for sUAS-based delivery systems?

Research Question 4 (RQ 4) : How can the small VTOL UAV design module and the VRP module be integrated with a FPI method to obtain a converged solution?

Thus, to address these research questions, this thesis will fulfill the following task

Research Objective (RO) : To develop a novel sizing and synthesis approach for small VTOL UAV that takes into account both planning vehicle operation on the flight network in which it will be operating and designing the flight network capable of addressing the obstructed environment as part of the vehicle design process.

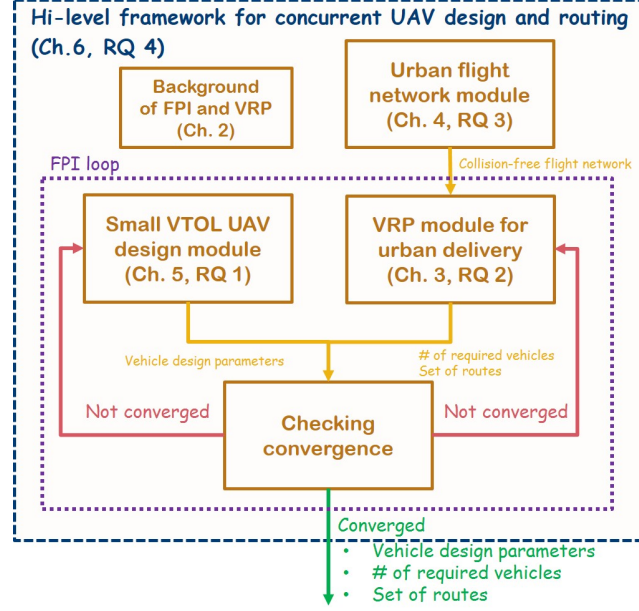


Figure 1.22: Structure of chapters of this thesis

In order to further examine specific ideas, Chapter 2 will address common knowledge of both the FPI theory and the VRP. Chapter 3 will address Research Question 2 (RQ 2) and present a mathematical optimization model of the UAS-based delivery system. Chapter 4 will discuss Research Question 3 (RQ 3) and extend the optimization model by feeding an urban flight network created by an actual urban model to address a collision-free path-planning method for UAVs in the urban environment. Chapter 5 will address Research Question 1 (RQ 1) and show a component-based sVTOL UAV sizing method. Finally, Chapter 6 will discuss Research Question 4 (RQ 4) and present a framework to solve a concurrent unmanned vehicle design and routing problem based on a FPI method as illustrated in Figure 1.22.

CHAPTER 2

BACKGROUND

This chapter contains background information on both the Fixed Point Iteration (FPI) method and the Vehicle Routing Problem (VRP) that are required to address the research questions described in the previous chapter. Section 2.1 presents mathematical background knowledge of a FPI method to deal with a Multi-Disciplinary Optimization (MDO) problem to address the research question such that

Research Question 4 (RQ 4) : How can the small VTOL UAV design module and the VRP module be integrated with a FPI method to obtain a converged solution?

Section 2.2 discusses background knowledge of the VRP and existing VRP models relevant to this thesis to address the research question such that

Research Question 2 (RQ 2) : What type of a VRP optimization model best represents the operations of sUAS-based delivery systems?

The intent of this chapter is to provide enough background on the FPI method and the VRP to unfamiliar readers to enable a comparison between theory and implementation or existing models and new models that will be presented in the following chapters.

2.1 Fixed Point Iteration (FPI) Method

This section presents basic FPI theory, and then extends its concept to a system of nonlinear equations to address an MDO problem. Lastly, this section presents a FPI method for a coupled problem as an MDO problem.

2.1.1 Fixed Point Iteration Theory

This Section presents a mathematical aspect to fixed point iteration theory based on [36, 42]. A nonlinear function of the form $f(x) = 0$ can be formulated as

$$x = f(x) \quad (2.1)$$

A solution x^* of Equation 2.1 is called a fixed point of the nonlinear equation. Furthermore, its existence can be proven. At first, let g be a continuous function on $[a, b]$. Then,

Theorem 1 (Brouwer's Fixed Point Theorem [42]). *If $g(x) \in [a, b]$ for each $x \in [a, b]$, then g has a fixed point in $[a, b]$.*

Proof. If $g(a) = a$ or $g(b) = b$, then g has a fixed point at a or b . If $g(a) \neq a$ and $g(b) \neq b$, then $g(a) > a$ and $g(b) < b$. Let us define a function, $f(x) = x - g(x)$, which is continuous on $[a, b]$. Then, $f(a) = a - g(a) < 0$, and $f(b) = b - g(b) > 0$. That is, $f(a)$ and $f(b)$ have opposite signs. There exists $c \in [a, b]$ such that $f(c) = 0$ due to the intermediate value theorem. Then, because $f(c) = c - g(c) = 0$, $c = g(c)$. Thus, c is a fixed point of $g(x)$, which completes the proof. \square

This theorem represents existence of the fixed point for a given range. To address uniqueness of the fixed point, additional condition [42] is required such that

Theorem 2. *If $g'(x) \in (a, b)$, $\forall x \in [a, b]$, and there exists a constant $0 < k < 1$ such that*

$$|g'(x)| \leq k, \quad \forall x \in (a, b)$$

then g has a unique fixed point x^ in $[a, b]$.*

Proof. Let us assume that $g(x)$ has two distinct fixed points, $c, d \in [a, b]$. Because $g'(x) \in (a, b)$, $g'(x)$ is differentiable in (a, b) . Then, by the mean value theorem, there exists $z \in$

(c, d) such that

$$g'(z) = \frac{g(c) - g(d)}{c - d} = \frac{c - d}{c - d} = 1$$

This implies $|g'(z)| = |1| = 1$. This contradicts the fact that $|g'(x)| \leq k \leq 1$, which completes the proof. \square

Thus, if a function satisfies Theorem 1 and 2, then existence and uniqueness of a fixed point is guaranteed. To address an iterative method, this mathematical concept of the fixed point of a function is required to be transferred to that of a sequence. A fixed point iteration of a sequence, which is also called the nonlinear Richardson iteration, is formulated by

$$x_{n+1} = f(x_n), \quad x = 1, 2, \dots \quad (2.2)$$

where n is an iteration index. The fixed point of the fixed point iteration is defined by the theorem [42] such that

Theorem 3. *if g is a continuous function and $\{x_n\}$ is a sequence generated by a fixed point iteration such that $\lim_{n \rightarrow \infty} x_n = x^*$, then x^* is a fixed point of g .*

Proof. Since $\lim_{n \rightarrow \infty} x_n = x^*$, $\lim_{n \rightarrow \infty} x_{n-1} = x^*$. Then, we have

$$\begin{aligned} x^* &= \lim_{n \rightarrow \infty} x_n \\ &= \lim_{n \rightarrow \infty} g(x_{n-1}) \\ &= g\left(\lim_{n \rightarrow \infty} x_{n-1}\right) \quad \because g \text{ is continuous} \\ &= g(x^*) \end{aligned}$$

Thus, x^* is a fixed point of g , which completes the proof. \square

However, not every sequence of the fixed point iteration converges, its convergence depends on derivative of the function. There are four types of convergence. First, if $0 < g'(x) < 1$, the sequence has monotone convergence. Second, if $-1 < g'(x) < 0$, the

sequence has oscillating convergence. Next, if $g'(x) > 1$, the sequence has monotone divergence. Last, if $g'(x) < -1$, the sequence has divergent oscillations.

To address the convergence of the fixed point iteration, Theorem 4 provides necessary conditions for a sequence to converge, which is written as

Theorem 4 (Contraction Mapping Theorem [42]). *Let g be a continuous function on the interval $[a, b]$. If $g(x) \in [a, b]$ for all each $x \in [a, b]$, and if there exists a constant $0 < L < 1$ such that*

$$|g(x) - g(y)| \leq L|x - y|, \quad x, y \in [a, b]$$

then g has a unique fixed point $x^ \in [a, b]$, and the sequence of iterates $\{x_k\}_{k=0}^{\infty}$ converges to x^* , for any initial guess $x_0 \in [a, b]$.*

Proof. Since $x_0 \in [a, b]$, $\{x_k\} \subseteq [a, b]$. If we take difference of successive iterations,

$$\begin{aligned} |x_{k+1} - x_k| &= |g(x_k) - g(x_{k-1})| \\ &\leq L|x_k - x_{k-1}| \leq \cdots \leq L^k|x_1 - x_0| \end{aligned}$$

$$\begin{aligned} |x_k - x_0| &= \left| \sum_{j=0}^{k-1} (x_{j+1} - x_j) \right| \\ &\leq \sum_{j=0}^{k-1} |x_{j+1} - x_j| \\ &\leq \sum_{j=0}^{k-1} L^j |x_1 - x_0| \\ &\leq \frac{1}{1-L} |x_1 - x_0| \end{aligned}$$

This implies $\{x_k\}$ is bounded.

$$\begin{aligned}
|x_{k+j} - x_k| &= |g(x_{k+j-1}) - g(x_{k-1})| \\
&\leq L|x_{k+j-1} - x_{k-1}| \\
&\leq L^2|x_{k+j-2} - x_{k-2}| \\
&\leq \cdots \leq L^k|x_j - x_0| \\
&\leq \frac{L^k}{1-L}|x_1 - x_0| \xrightarrow{k \rightarrow \infty} 0
\end{aligned}$$

This implies that $\{x_k\}$ is Cauchy. That is, it has a limit. Let the limit be x^* , then $x^* = g(x^*)$.

In order to prove its uniqueness, let x^*, y^* be such that $g(x^*) = x^*$, and $g(y^*) = y^*$, then

$$|x^* - y^*| = |g(x^*) - g(y^*)| \leq L|x^* - y^*|$$

This implies that $x^* = y^*$. Furthermore,

$$\begin{aligned}
|x_{k+1} - x^*| &= |g(x_k) - g(x^*)| \\
&\leq L|x_k - x^*|
\end{aligned}$$

Since $0 < L < 1$, $\{x_k\}$ converges to x^* , which completes the proof. \square

The results of Theorem 4 based on the mapping theory lead to another theorem based on the derivative of a function such that

Theorem 5 (Fixed Point Theorem [42]). *Let g be a continuous function on the interval $[a, b]$, and let g be differentiable on (a, b) . If $g(x) \in (a, b)$ for each $x \in [a, b]$, and if there exists a constant $0 < L < 1$ such that*

$$|g'(x)| \leq L, \quad x \in (a, b)$$

then the sequence of iterates $\{x_k\}_{k=0}^{\infty}$ converges to the unique fixed point x^* of g in $[a, b]$, for any initial guess $x_0 \in [a, b]$.

Proof. Since $x_0 \in [a, b]$, $\{x_k\} \subseteq [a, b]$. By the mean value theorem, there exists $c \in (a, c)$ such that

$$\begin{aligned}
 g'(c) &= \frac{g(x_k) - g(x^*)}{x_k - x^*} \\
 |x_k - x^*| &= |g'(c)| |g(x_k) - g(x^*)| \\
 &\leq L |g(x_k) - g(x^*)| \quad \because |g'(x)| \leq L \\
 &\leq L |x_{k-1} - x^*| \\
 &\leq L^2 |x_{k-2} - x^*| \\
 &\leq \dots \leq L^k |x_0 - x^*|
 \end{aligned} \tag{2.3}$$

Since $\lim_{k \rightarrow \infty} L^k = 0$ for $0 < L < 1$,

$$\lim_{k \rightarrow \infty} |x_k - x^*| = 0$$

This implies that x_k converges to x^* , which completes the proof. \square

This subsection presented the mathematical background of the fixed point of a function and a sequence to address existence and uniqueness of the fixed point.

2.1.2 Fixed Point Iteration for a System of Nonlinear Equations

Based on Theorems in Section 2.1.1, the fixed point iteration method can be generalized for a problem to solve a system of nonlinear equations [42]. In order to solve $\mathbf{Ax} = \mathbf{b}$, the fixed point iteration can be written such that

$$\mathbf{x}^{k+1} = \mathbf{T}\mathbf{x}^k + \mathbf{M}^{-1}\mathbf{b} \tag{2.4}$$

where $\mathbf{A} = \mathbf{M} - \mathbf{N}$, $\mathbf{T} = \mathbf{M}^{-1}\mathbf{N}$ which is an iteration matrix. Then, let us consider a system of nonlinear equations such that

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \dots, x_n) &= 0 \end{aligned}$$

As a matrix form, the problem can be written by $\mathbf{F}(\mathbf{x}) = 0$. Then, this system can be transformed to

$$\mathbf{x} = \mathbf{G}(\mathbf{x}) \tag{2.5}$$

Furthermore, with an initial guess $\mathbf{x}^{(0)}$, the fixed point iteration can be defined by

$$\mathbf{x}^{(k+1)} = \mathbf{G}(\mathbf{x}^{(k)}), \quad k = 0, 1, 2, \dots \tag{2.6}$$

The existence of uniqueness of the fixed point iteration of either Equation (2.5) or (2.6) can be proved in an analogous manner from the single-variable case in Chapter 2.1.1. Let us consider a domain $\mathbf{D} \subseteq \mathbb{R}^n$ and a function \mathbf{G} which maps \mathbf{D} into \mathbf{D} . If \mathbf{G} is continuous on \mathbf{D} , then there exists a fixed point of \mathbf{G} . Furthermore, if the partial derivatives of \mathbf{G} are continuous on \mathbf{D} and there exists a constant $\rho < 1$ such that

$$\|\mathbf{J}_{\mathbf{G}}(\mathbf{x})\| \leq \rho, \quad \mathbf{x} \in \mathbf{D} \tag{2.7}$$

where $\mathbf{J}_{\mathbf{G}}(\mathbf{x})$ is the Jacobian matrix of \mathbf{G} , then \mathbf{G} has a unique fixed point \mathbf{x}^* in \mathbf{D} for any initial guess $\mathbf{x}^{(0)}$.

With these generalized concepts of the fixed point iteration, the FPI method has successfully solved MDO problems. This paper implements a fixed point iteration method

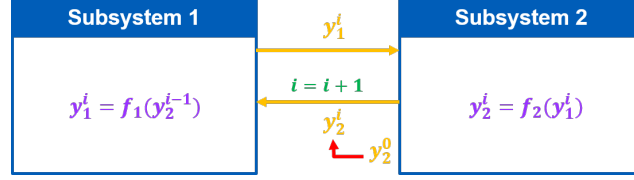


Figure 2.1: FPI for a system consisting of two nonlinear equations

for the concurrent UAV design and vehicle routing problems. In next Section, engineering aspects of the FPI method for a practical application will be addressed.

2.1.3 Fixed Point Iteration Method

Now, let us consider a system consisting of two subsystems which are coupled each other as illustrated in Figure 2.1. In order to obtain a converged solution of the FPI, an initial guess y_2^0 is required. Based on y_2^0 , the FPI process starts, and iterates until convergence criteria are satisfied. Detailed steps of the FPI are described in Algorithm 1 [42].

Algorithm 1 Pseudo algorithm for FPI [42]

- Step 0: Choose initial value y_2^0 , set $i = 0$, and set *maxiter* and ε
 - Step 1: $i = i + 1$
 - Step 2: $y_1^i = f_1(y_2^{i-1})$
 - Step 3: $y_2^i = f_2(y_1^i)$
 - Step 4: if $|y_1^i - y_2^i| < \varepsilon$ then stop, otherwise continue
 - Step 5: if $i \geq \text{maxiter}$ then stop, otherwise go to step 1
-

The FPI tries to find a converged solution of a system of equations, but it has no functions to find a better solution when there are multiple converged solutions. Thus, when the FPI is used to solve a Multi-Disciplinary Optimization (MDO) problem, different initial guesses may be used to find a better solution. That is, the FPI can find a local optimum of an MDO problem.

This section discussed mathematical background of the fixed point theory and the fixed point method because this thesis uses a fixed point iteration method to find a converged solution of integrated module with both a sUAV design module and a VRP module for

urban delivery as stated previously. Chapter 6 presents more details of implementation of the FPI method.

2.2 Vehicle Routing Problem (VRP)

This section addresses the Vehicle Routing Problem (VRP). At first, Subsection 2.2.1 ~ 2.2.4 present fundamental concepts for the vehicle routing problem such as the definition of a graph and an integer programming problem. Then, Subsection 2.2.5 discusses a traveling salesman problem, which is a special case of a vehicle routing problem having a single vehicle. Finally, Subsection 2.2.6 addresses the vehicle routing problem with existing VRP models. The readers interested in vehicle routing problems may refer Appendix A which presents practical vehicle routing problems for unmanned aircraft systems with approaches to solving the given problems.

2.2.1 Graph Theory

The vehicle routing problem requires an input describing a mission area. For a VRP of delivery systems, the input should include a set of locations of depots and delivery points with connectivity information for each pair of two locations. In graph theory which is a study field in discrete mathematics, a graph can naturally represent the connectivity information of a set of locations. A mathematical definition of graph is “A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a finite, nonempty set $\mathcal{V} = \{1, 2, \dots, m\}$ and a set $\mathcal{E} = \{e_1, e_2, \dots, e_n\}$ whose elements are subsets of \mathcal{V} of size 2, that is, $e_k = (i, j)$, where $i, j \in \mathcal{V}$. The elements of \mathcal{V} are called nodes, and the elements of \mathcal{E} are called edges. Thus graphs are a mechanism for specifying certain pairs of a set [43].” In vehicle routing problems, the entire mission area is translated to a graph.

For instance, a symmetric flight network of homogeneous UAVs for a delivery mission is able to be modeled as a graph similar to Figure 2.2. Mission points and flight paths are translated to nodes and edges in a graph respectively. Edges are possible options of flight

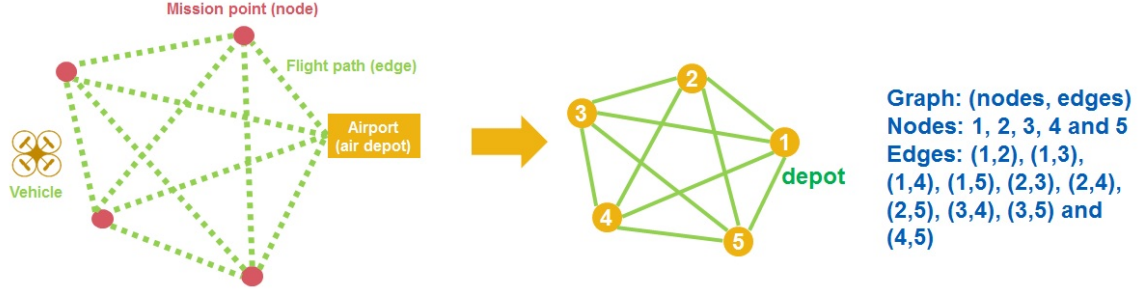


Figure 2.2: Modeling a flight network by a graph

paths, and they can have attributes such as flight distance and flight time.

If the values of attributes are the same regardless of the direction of edges, then the network is called a symmetric network/graph. Otherwise, the network is called an asymmetric network/graph. The VRP optimization model is normally formulated based on an asymmetric network/graph.

Based on the network, a given mission and its concept of operations can be formulated as a combinatorial optimization problem, which will be discussed in the following subsection, with constraints such that all mission points should be visited exactly once for a delivery mission, or each route of vehicles should start from and end at the depot. Note that the constraints in the problem depends on its input graph because the graph includes all possible options. For instance, all routes as a solution of the delivery mission example is a subset of input graph of the problem. That is, a graph provides a domain of design space of a VRP. That is why this section start to discuss a graph to present the vehicle routing problem.

2.2.2 Combinatorial Optimization Problem

Given a finite set $\mathcal{N} = \{1, \dots, n\}$, weights c_j for each $j \in \mathcal{N}$, and a set \mathcal{F} consisting of feasible subsets of \mathcal{N} , a combinatorial optimization problem is defined as

$$\min_{\mathbf{S} \subseteq \mathcal{N}} \left\{ \sum_{j \in \mathbf{S}} c_j : \mathbf{S} \in \mathcal{F} \right\} \quad (2.8)$$

Furthermore, given a finite set $\mathcal{N} = \{1, \dots, n\}$, vectorial weights $w : \mathcal{N} \rightarrow \mathbb{Q}^d$, convex functional $c : \mathbb{Q}^d \rightarrow \mathbb{Q}$, and a set \mathcal{F} of feasible subsets of \mathcal{N} , a Convex Combinatorial Optimization (CCO) problem is defined as [44]

$$\min c(w(\mathbf{F})), \quad \mathbf{F} \in \mathcal{F} \quad (2.9)$$

Since a linear function is a convex function, a Linear Combinatorial Optimization (LCO) problem is a CCO problem. Typical applications of the combinatorial optimizations are the shortest path problem, the Traveling Salesman Problem (TSP), and the Vehicle Routing Problem (VRP). In the delivery example as stated previously, a solution of the problem is a feasible combination of edges. Thus, the delivery problem can be formulated by a CCO problem.

2.2.3 Integer Programming (IP)

In order to solve a combinatorial optimization problem, the problem can be formulated as an integer programming problem [45]. This allows to use a variety of techniques and methods which have been researched and developed for linear/integer programming problems such as a simplex method and a relaxation technique [46].

At first, given \mathbf{A} is an $m \times n$ matrix, \mathbf{c} is a $1 \times n$ vector and \mathbf{b} is an $m \times 1$ vector, a Linear Programming (LP) problem is formulated as

$$\begin{aligned} \max \quad & \mathbf{c}\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq 0 \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned} \quad (2.10)$$

An integer programming problem is a special case of a linear programming problem.

Given \mathbf{A} is an $m \times n$ matrix, \mathbf{c} is a $1 \times n$ vector and \mathbf{b} is an $m \times 1$ vector, an integer programming problem is defined as

$$\begin{aligned}
& \max \quad \mathbf{c}\mathbf{x} \\
& s.t. \quad \mathbf{A}\mathbf{x} \leq \mathbf{b} \\
& \quad \mathbf{x} \geq 0 \\
& \quad \mathbf{x} \in \mathbb{Z}^n
\end{aligned} \tag{2.11}$$

In order to use techniques solving a linear programming problem to solve an integer programming problem, a linear relaxation method can be used [46]. A linear programming problem is used as a subproblem when solving an integer programming problem. Thus, an integer programming problem is significantly more computationally expensive to solve than a linear programming problem.

If all design variables are restricted to binary values, the integer programming problem can be translated to a 0-1 or binary integer programming problem defined as

$$\begin{aligned}
& \max \quad \mathbf{c}\mathbf{x} \\
& s.t. \quad \mathbf{A}\mathbf{x} \leq \mathbf{b} \\
& \quad \mathbf{x} \in \{0, 1\}^n
\end{aligned} \tag{2.12}$$

Since a constraint involving binary variables represents disjunct conditions itself, the big-M relaxation can be used [46]. In general, the big-M relaxation translates a logic or non-convex constraints into other constraints with auxiliary binary variables describing the same feasible space. For constraints including binary variables only, the big-M relaxation can express on/off condition for each binary variable. The readers interested in the big-M relaxation may refer [46].

Finally, a mixed integer programming problem including n design variables and m

constraints can be defined as

$$\begin{aligned}
& \max && \mathbf{c}^T \mathbf{x} \\
& s.t. && \mathbf{A} \mathbf{x} \leq \mathbf{b} \\
& && \mathbf{x} \geq 0 \\
& && \mathbf{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p}
\end{aligned} \tag{2.13}$$

where \mathbf{c}^T is a cost coefficient vector, $\mathbf{c}^T \in \mathbb{R}^n$, and \mathbf{A}/\mathbf{b} are a constraint matrix/vector, $\mathbf{A} \in \mathbb{R}^{n \times m}$ and $\mathbf{b} \in \mathbb{R}^m$, and \mathbf{x} is a design variable vector consisting of p integer numbers and $n - p$ real numbers.

Mixed integer programming has been used to solve a wide variety of practical problems such as train scheduling, airline crew scheduling, production planning, electricity generation planning, telecommunications, buses for the handicapped, ground holding of aircraft and cutting problems [47]. This thesis formulates a vehicle routing problem as a mixed integer programming problem because each sUAV's flight distance/time is captured by real/continuous variables while each route of sUAVs is combination of selected edges described by binary variables.

2.2.4 Complexity

A combinatorial optimization problem can be translated to a decision problem which is a yes-or-no question. At first, the polynomial time (\mathcal{P}) class of decision problems can be solved in polynomial time by a deterministic Turing machine defined by the 7-tuple such that [48]

$$DTM = (Q, \Sigma, \Gamma, \delta, q_0, B, F) \tag{2.14}$$

where Q is the finite set of states of the finite control, and Σ is the finite set of input symbols, and Γ is the complete set of tape symbols such that $\Sigma \subset \Gamma$, and q_0 is the start state such that $q_0 \in Q$, B is the blank symbol such that $B \in \Gamma$ and $B \notin \Sigma$, and F is the set of final

or accepting states such that $F \subset Q$, and δ is the transition function which determines the next behavior, state, and move direction such that

$$\delta(q, X) \Longrightarrow (p, Y, D) \quad (2.15)$$

where q is the current state, and X is the tape symbol, and p is the next state such that $p \in Q$, and Y is the symbol which will be written in the cell of the tape such that $Y \in \Gamma$, and D is the next move direction such as left or right.

Second, the Nondeterministic Polynomial time (\mathcal{NP}) class of decision problems can be solved in polynomial time by a nondeterministic Turing machine that can pursue an unbounded number of independent computational sequences in parallel [49]. A nondeterministic Turing machine also is defined by the 7-tuple, like a deterministic Turing machine, such that

$$NTM = (Q, \Sigma, \Gamma, \delta, q_0, B, F) \quad (2.16)$$

The difference is that the transition function δ of a nondeterministic Turing machine has multiple options such that

$$\delta(q, X) \Longrightarrow \{(p_1, Y_1, D_1), (p_2, Y_2, D_2), \dots, (p_k, Y_k, D_k)\} \quad (2.17)$$

where k is any finite integer [48]. Alternatively, a decision problem is in the \mathcal{NP} class if the problem has an efficient certifier which can verify a solution in polynomial time [50]. Hence, the definition of the \mathcal{P} class focuses on the solvability of a problem whereas that of the \mathcal{NP} class concentrates on the provability of a solution of a problem. By the alternative definition, a relation between the \mathcal{P} class and the \mathcal{NP} class is observed [48] such that

$$\mathcal{P} \subseteq \mathcal{NP} \quad (2.18)$$

Next, a decision problem in the \mathcal{NP} class is in the \mathcal{NP} -complete class if the problem

can be reduced to another problem in the \mathcal{NP} class in polynomial time, where reducing is a mapping from a problem to another problem [49]. Cook [51] proves that a satisfiability problem which represents a circuit consisting of Boolean operators is an \mathcal{NP} -complete problem. Then, Karp [52] proves that the 21 decision problems of combinatorial optimization problems can be reduced to the satisfiability problem. A directed/undirected Hamilton circuit problem which is a simple version of a traveling salesman problem is included the 21 problems.

Finally, a decision problem is in the \mathcal{NP} -hard class if a \mathcal{NP} -complete problem can be reduced to the decision problem [53]. A \mathcal{NP} -hard problem needs not to be in the \mathcal{NP} class. Alternatively, there is an informal definition such that a decision problem is in the \mathcal{NP} -hard class if the problem is at least as hard as the \mathcal{NP} -complete class. Note that when mentioning the complexity of an optimization problem, the \mathcal{NP} -hard class is referred instead of the \mathcal{NP} -complete class because technically the \mathcal{NP} -complete class applies to decision problems only [50]. Thus, a vehicle routing problem addressed in this thesis, which is a generalized traveling salesman problem, is an \mathcal{NP} -hard problem. Since the more precise definitions of the complexity are based on a language instead of a problem, the readers interested in the complexity can refer [49, 51, 52].

2.2.5 Existing Traveling Salesman Problem (TSP)

A Traveling Salesman Problem (TSP) is to find the minimum trip cost to visit a list of cities with two conditions; one is that a salesman can visit a city exactly once, and the other is that the salesman should return to the home city [54, 55, 56]. Based on graph theory, a simple version of a traveling salesman problem is able to be transformed to a Hamiltonian cycle problem which determines a closed loop visiting every vertex in a graph exactly once [57]. Since a directed/undirected Hamilton circuit problem is in the \mathcal{NP} -complete class, a traveling salesman problem is an \mathcal{NP} -hard problem.

For instance, a traveling salesman problem can be applied to a single-UAV patrol mis-

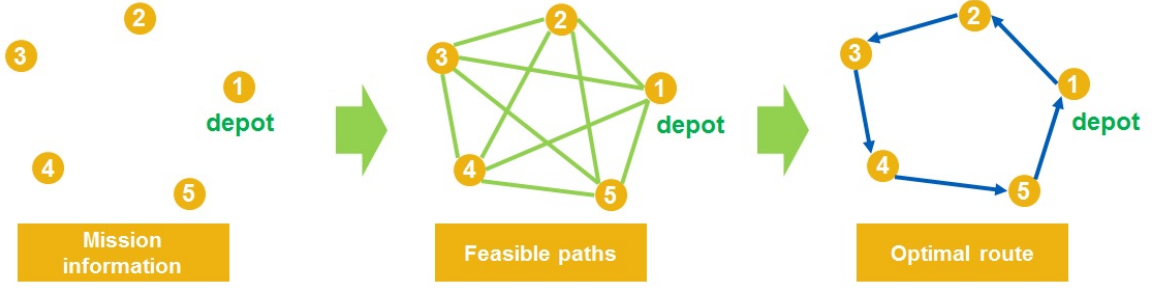


Figure 2.3: Example of a traveling salesman problem

sion planning. Then, the home city in a traveling salesman problem can be translated to a depot in a flight network of the mission area. By solving a traveling salesman problem, the optimal route visiting all mission points exactly once and returning to the depot is determined as illustrated in Figure 2.3.

Hoffman et al. [56] introduced a common mathematical form of a traveling salesman problem. To formulate a traveling salesman problem, binary design variables are defined as

$$x_{ij} = \begin{cases} 1 & \text{if the edge from } i \text{ to } j \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \quad (2.19)$$

Given a list of cities, $\mathcal{V} = \{1, \dots, m\}$, and any nonempty proper subset, \mathcal{K} , of the list of cities, the traveling salesman problem is formulated by the binary design variables such that

$$\min \sum_{j=1}^m \sum_{i=1}^m c_{ij} x_{ij} \quad (2.20)$$

$$s.t. \quad \sum_{j=1}^m x_{ij} = 1 \quad (\forall i \in \mathcal{V}) \quad (2.21)$$

$$\sum_{i=1}^m x_{ij} = 1 \quad (\forall j \in \mathcal{V}) \quad (2.22)$$

$$\sum_{i \in \mathcal{K}} \sum_{j \in \mathcal{K}} x_{ij} \leq |\mathcal{K}| - 1 \quad (\mathcal{K} \subset \mathcal{V}) \quad (2.23)$$

$$x_{ij} \in \{0, 1\} \quad (\forall i, j \in \mathcal{V}) \quad (2.24)$$

where c_{ij} is cost of the edge (i, j) . Equation (2.21) and (2.22) are inbound and outbound

constraints for each city. Equation (2.23) removes subtours which create disjoint loops. This is a common mathematical model for the Asymmetric Traveling Salesman Problem (ATSP).

For an undirected graph, a traveling salesman problem can be reformulated with new binary design variables defined as

$$x_j = \begin{cases} 1 & \text{if the edge } j \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \quad (2.25)$$

Let \mathcal{E} be all edges and c_k be cost of edge k . With the new design variables, the Symmetric Traveling Salesman Problem (STSP) can be formulated as

$$\min \frac{1}{2} \sum_{j=1}^m \sum_{k \in \mathcal{J}(j)} c_k x_k \quad (2.26)$$

$$s.t. \quad \sum_{k \in \mathcal{J}(j)} x_k = 2 \quad (\forall j \in \mathcal{V}) \quad (2.27)$$

$$\sum_{j \in \mathcal{E}(\mathcal{K})} x_j \leq |\mathcal{K}| - 1 \quad (\mathcal{K} \subset \mathcal{V}) \quad (2.28)$$

$$x_j \in \{0, 1\} \quad (\forall j \in \mathcal{E}) \quad (2.29)$$

where $\mathcal{J}(j)$ is the set of all undirected edges connected to node j and $\mathcal{E}(\mathcal{K})$ is the subset of all undirected edges connecting the cities in any nonempty proper subset, \mathcal{K} , of all cities. Note that although the Symmetric Traveling Salesman Problem (STSP) is a special case of the Asymmetric Traveling Salesman Problem (ATSP), an algorithm for a ATSP has performed badly on a STSP. Thus, algorithms for solving a STSP had researched independently [56]. However, the VRP normally addresses asymmetric network.

The TSP can be generalized by considering several salesmen instead of a single salesman. The extension is the VRP which intrinsically consider multiple vehicles. Thus, this thesis models the operations of sUAS-based delivery systems as a VRP rather than a TSP.

2.2.6 Existing Vehicle Routing Problem (VRP) Models

According to Toth and Vogo [58], a Vehicle Routing Problem (VRP) is defined as a problem which finds a plan determining a set of vehicle routes to perform all given logistics requests with the given fleet of vehicles at minimum cost. A vehicle routing problem was first proposed by Dantzig and Ramser [59] as a truck dispatching problem. Clark and Wright [60] present a greedy heuristic method called savings algorithm to address a vehicle routing problem. Furthermore, during the past five decades, a variety of variants of vehicle routing problem as shown in Figure 2.4 have been studied with more and more practical constraints and hundreds of papers have been devoted to the Capacitated Vehicle Routing Problem (CVRP), which is the basic model of the VRP using multiple vehicles having limited payload capacity, or the Vehicle Routing Problem with Time Windows (VRPTW) which addresses the on-demand delivery concept based on the CVRP [61].

In this sense, this section addresses mathematical models for a Capacitated Vehicle Routing Problem (CVRP) and a Vehicle Routing Problem with Time Windows (VRPTW). As mentioned previously, the operations of sUAS-based delivery systems should consider reuse of sUAVs due to short endurance and small payload capacity. To address these properties, the concept of a Multi-Trip Vehicle Routing Problem with Time Windows (MTVRPTW) is introduced as a relevant variant of this thesis with a notional example. A multi-trip vehicle routing problem with time windows is fully addressed in Section 3.2. Some practical applications for multiple UAVs are presented in Appendix A.

Capacitated Vehicle Routing Problem (CVRP)

A Capacitated Vehicle Routing Problem (CVRP) is a fundamental variant of vehicle routing problems. Specifically, the problem is to find a plan that a fleet of vehicles transports all packages to customers at minimum cost with the constraints as follows

- Each customer should be visited exactly once by a vehicle

1. Type of Study	2.8. Backhauls	3.9. Vehicle homogeneity (Capacity)
1.1. Theory	2.8.1. Nodes request simultaneous pick ups and deliveries	3.9.1. Similar vehicles
1.2. Applied methods	2.8.2. Nodes request either linehaul or backhaul service, but not both	3.9.2. Load-specific vehicles ²
1.2.1. Exact methods	2.9. Node/Arc covering constraints	3.9.3. Heterogeneous vehicles
1.2.2. Heuristics	2.9.1. Precedence and coupling constraints	3.9.4. Customer-specific vehicles ³
1.2.3. Simulation	2.9.2. Subset covering constraints	3.10. Travel time
1.2.4. Real time solution methods	2.9.3. Re course allowed	3.10.1. Deterministic
1.3. Implementation documented	3. Problem Physical Characteristics	3.10.2. Function dependent (a function of current time)
1.4. Survey, review or meta-research	3.1. Transportation network design	3.10.3. Stochastic
2. Scenario Characteristics	3.1.1. Directed network	3.10.4. Unknown
2.1. Number of stops on route	3.1.2. Undirected network	3.11. Transportation cost
2.1.1. Known (deterministic)	3.2. Location of addresses (customers)	3.11.1. Travel time dependent
2.1.2. Partially known, partially probabilistic	3.2.1. Customers on nodes	3.11.2. Distance dependent
2.2. Load splitting constraint	3.2.2. Arc routing instances	3.11.3. Vehicle dependent ⁴
2.2.1. Splitting allowed	3.3. Geographical location of customers	3.11.4. Operation dependent
2.2.2. Splitting not allowed	3.3.1. Urban (scattered with a pattern)	3.11.5. Function of lateness
2.3. Customer service demand quantity	3.3.2. Rural (randomly scattered)	3.11.6. Implied hazard/risk related
2.3.1. Deterministic	3.3.3. Mixed	4. Information Characteristics
2.3.2. Stochastic	3.4. Number of points of origin	4.1. Evolution of information
2.3.3. Unknown ¹	3.4.1. Single origin	4.1.1. Static
2.4. Request times of new customers	3.4.2. Multiple origins	4.1.2. Partially dynamic
2.4.1. Deterministic	3.5. Number of points of loading/unloading facilities (depot)	4.2. Quality of information
2.4.2. Stochastic	3.5.1. Single depot	4.2.1. Known (Deterministic)
2.4.3. Unknown	3.5.2. Multiple depots	4.2.2. Stochastic
2.5. On site service/waiting times	3.6. Time window type	4.2.3. Forecast
2.5.1. Deterministic	3.6.1. Restriction on customers	4.2.4. Unknown (Real-time)
2.5.2. Time dependent	3.6.2. Restriction on roads	4.3. Availability of information
2.5.3. Vehicle type dependent	3.6.3. Restriction on depot/hubs	4.3.1. Local
2.5.4. Stochastic	3.6.4. Restriction on drivers/vehicle	4.3.2. Global
2.5.5. Unknown	3.7. Number of vehicles	4.4. Processing of information
2.6. Time window structure	3.7.1. Exactly n vehicles (<i>TSP in this segment</i>)	4.4.1. Centralized
2.6.1. Soft time windows	3.7.2. Up to n vehicles	4.4.2. Decentralized
2.6.2. Strict time windows	3.7.3. Unlimited number of vehicles	5. Data Characteristics
2.6.3. Mix of both	3.8. Capacity consideration	5.1. Data Used
2.7. Time horizon	3.8.1. Capacitated vehicles	5.1.1. Real world data
2.7.1. Single period	3.8.2. Uncapacitated vehicles	5.1.2. Synthetic data
2.7.2. Multi period		5.1.3. Both real and synthetic data
		5.2. No data used

¹Unknown refers to the case in which information is revealed in real-time (i.e., dynamic and fuzzy studies fall under this category)

²Each vehicle can be used to handle specific types of loads

³A customer must be visited by a specific type of vehicle

⁴Cost of operating a vehicle is not negligible

Figure 2.4: Taxonomy of the VRP literature [25]

- Each route starts/ends at a depot
- Each vehicle can carry packages up to its maximum package capacity

A notional example of a CVRP is illustrated in Figure 2.5. For mathematical formulations, let \mathcal{N} and \mathcal{A} be all nodes and arcs respectively. Furthermore, \mathcal{S} is a subset of \mathcal{V} such that $\mathcal{S} \subseteq \mathcal{N}$, and $r(\mathcal{S})$ is the minimum number of vehicles to transport all packages to customers. Before solving a capacitated vehicle routing problem, the minimum number of vehicles $r(\mathcal{S})$ should be determined. In order to find $r(\mathcal{S})$, a bin packing problem having n

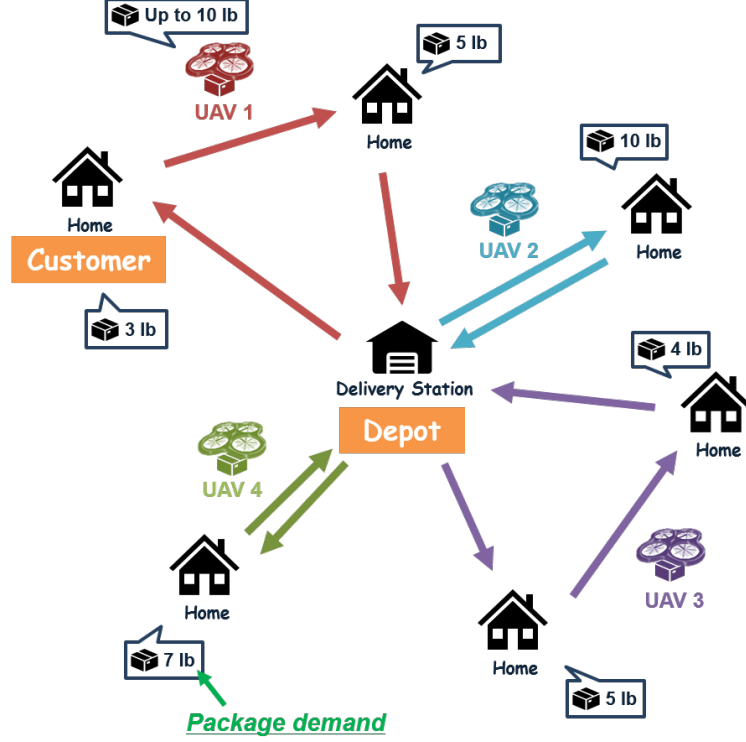


Figure 2.5: Notional example of a Vehicle Routing Problem (VRP)

packages and n knapsacks/bins/vehicles can be used, which is defined as [62]

$$\min \sum_{i=1}^n y_i \quad (2.30)$$

$$s.t. \quad \sum_{j=1}^n w_j x_{ij} \leq c y_i \quad \text{for } i \in \mathcal{N} = \{1, \dots, n\} \quad (2.31)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad \text{for } \forall j \in \mathcal{N} \quad (2.32)$$

$$w_j \leq c \quad \text{for } j \in \mathcal{N} \quad (2.33)$$

$$c \in \mathbb{Z}^+ \quad (2.34)$$

$$y_i \in \{0, 1\} \quad \text{for } i \in \mathcal{N} \quad (2.35)$$

$$x_{ij} \in \{0, 1\} \quad \text{for } i, j \in \mathcal{N} \quad (2.36)$$

where c is the maximum package capacity, w_j is the package weight of customer j , and the

design variables are defined as

$$y_i = \begin{cases} 1 & \text{if the vehicle } i \text{ is used} \\ 0 & \text{otherwise} \end{cases} \quad (2.37)$$

$$x_{ij} = \begin{cases} 1 & \text{if the package } i \text{ is assigned to the vehicle } j \\ 0 & \text{otherwise} \end{cases} \quad (2.38)$$

Then, each vehicle can carry packages up to its package capacity by Equation (2.31), and each package should be assigned to a vehicle by Equation (2.32).

After finding $r(\mathcal{S})$, a capacitated vehicle routing problem can be addressed with the design variables defined as

$$x_{ij} = \begin{cases} 1 & \text{if the edge } (i,j) \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \quad (2.39)$$

Then, a capacitated vehicle routing problem can be formulated such that [58]

$$\min \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij} \quad (2.40)$$

$$s.t. \quad \sum_{j \in \delta^+(i)} x_{ij} = 1 \quad \text{for } \forall i \in \mathcal{N} \quad (2.41)$$

$$\sum_{i \in \delta^-(j)} x_{ij} = 1 \quad \text{for } \forall j \in \mathcal{N} \quad (2.42)$$

$$\sum_{j \in \delta^+(0)} x_{0j} \leq |K| \quad (2.43)$$

$$\sum_{(i,j) \in \delta^+(\mathcal{S})} x_{ij} \geq r(\mathcal{S}) \quad \text{for } \forall \mathcal{S} \subset \mathcal{N}, \mathcal{S} \neq \emptyset \quad (2.44)$$

$$x_{ij} \in \{0, 1\} \quad \text{for } \forall (i, j) \in \mathcal{A} \quad (2.45)$$

where K is the number of vehicles, $\delta^-(\mathcal{S})$ is the in-arcs such that $\delta^-(\mathcal{S}) = \{(i, j) \in \mathcal{A} : i \notin \mathcal{S}, j \in \mathcal{S}\}$, and $\delta^+(\mathcal{S})$ is the out-arcs such that $\delta^+(\mathcal{S}) = \{(i, j) \in \mathcal{A} : i \in \mathcal{S}, j \notin \mathcal{S}\}$. Then, flow balance at each node is guaranteed by Equation (2.41) and (2.42). The maximum number of required vehicles is restricted by Equation (2.43). Finally, Equation (2.44) serves

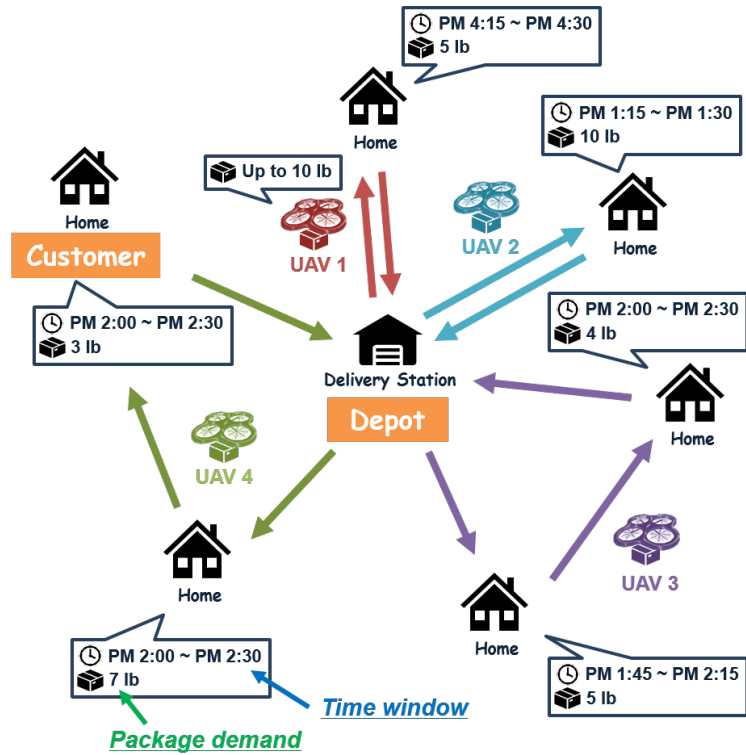


Figure 2.6: Notional example of a Vehicle Routing Problem with Time Windows (VRPTW)

as both capacity constraints and subtour elimination constraints. The other variants of vehicle routing problems add additional constraints to address a more detailed concept of operations for a delivery system.

Vehicle Routing Problem with Time Windows (VRPTW)

A Vehicle Routing Problem with Time Windows (VRPTW) can address an on-demand delivery system by adding time windows representing available time for delivery. The fundamental assumption of time window constraints is that a customer allows to be visited within available time period only. Specifically, a vehicle routing problem with time windows addresses the constraints such that

- Each customer should be visited exactly once by a vehicle
- Each route starts/ends at a depot
- Each vehicle can carry packages up to its maximum package capacity

- Vehicles can delivery packages to customers within each time window only
- If a vehicle arrives at a customer earlier than customer's time window, it should wait until open time of the time window (on-demand delivery)

A notional example of a vehicle routing problem with time windows is illustrated in Figure 2.6. In this example, each customer has a time window presenting avail time to be visited. To address mathematical formulations of a vehicle routing problem with time windows, let $\mathcal{N} = \{0, 1, \dots, n, (n+1)\}$, $\mathcal{C} = \{1, \dots, n\}$, and $\mathcal{V} = \{1, \dots, m\}$ be all nodes, all customers such that $\mathcal{C} \cup \{0, (n+1)\} = \mathcal{N}$, and a fleet of vehicles, respectively. Furthermore, d_i is the demand of customer i , q is the capacity for each vehicle, c_{ij} is moving cost from customer i to customer j , t_{ij} is moving time from customer i to customer j , and $[a_i, b_i]$ is the time window of customer i . Note that node 0 and $(n+1)$ represent a single depot, they are used to address the starting/ending point of a route. Then, a vehicle routing problem with time windows can be defined as [63]

$$\min \sum_{k \in \mathcal{V}} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} c_{ij} x_{ijk} \quad (2.46)$$

$$s.t. \quad \sum_{k \in \mathcal{V}} \sum_{j \in \mathcal{N}} x_{ijk} = 1 \quad \forall i \in \mathcal{C} \quad (2.47)$$

$$\sum_{i \in \mathcal{C}} d_i \sum_{j \in \mathcal{N}} x_{ijk} \leq q \quad \forall k \in \mathcal{V} \quad (2.48)$$

$$\sum_{j \in \mathcal{N}} x_{0jk} = 1 \quad \forall k \in \mathcal{V} \quad (2.49)$$

$$\sum_{i \in \mathcal{N}} x_{ihk} - \sum_{j \in \mathcal{N}} x_{hjk} = 0 \quad \forall h \in \mathcal{C}, \forall k \in \mathcal{V} \quad (2.50)$$

$$\sum_{i \in \mathcal{N}} x_{i,(n+1),k} = 1 \quad \forall k \in \mathcal{V} \quad (2.51)$$

$$x_{ijk}(s_{ik} + t_{ij} - s_{jk}) \leq 0 \quad \forall i, j \in \mathcal{N}, \forall k \in \mathcal{V} \quad (2.52)$$

$$a_i \leq s_{ik} \leq b_i \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{V} \quad (2.53)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i, j \in \mathcal{N}, \forall k \in \mathcal{V} \quad (2.54)$$

where x_{ijk} and s_{ik} are design variables. For each arc (i, j) , where $i \neq j$, $i \neq (n+1)$, $j \neq 0$,

x_{ijk} is defined as

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ drives directly from vertex } i \text{ to vertex } j \\ 0 & \text{otherwise} \end{cases} \quad (2.55)$$

Furthermore, s_{ik} is defined as the time when vehicle k starts to service customer i . Since x_{ijk} is a binary variable and s_{ik} is a continuous variable, a vehicle routing problem with time windows is a Mixed Integer Programming (MIP) problem.

In a solution of a vehicle routing problem with time windows, each customer is visited exactly once by Equation (2.47). Each vehicle can be loaded up to its capacity by Equation (2.48). Equation (2.49), (2.50) and (2.51) guarantee that each vehicle must leave and arrive at the depot. The relationship between the vehicle departure time from a customer and its immediate successor such that $s_{ik} < s_{jk}$ if $i < j$ and $t_{ij} > 0$ is maintained by Equation (2.52). Finally, Equation (2.53) represents time window constraints.

Multi-trip Vehicle Routing Problem with Time Windows (MTVRPTW)

A Multi-Trip Vehicle Routing Problem with Time Windows (MTVRPTW) can additionally address reuse of vehicles based on a VRPTW. The fundamental assumption for reuse of vehicles of this model is that processes for reuse of vehicles, such as reloading packages and refueling vehicles, is executed at the depot only. Specifically, a multi-trip vehicle routing problem with time windows is to build a plan that a fleet of vehicles transports all packages to customers at minimum cost with the constraints as follows

- Each customer should be visited exactly once by a vehicle
- Each route starts/ends at a depot
- Each vehicle can carry packages up to its maximum package capacity
- Vehicles can delivery packages to customers within each time window only
- If a vehicle arrives at a customer earlier than customer's time window, it should wait until open time of the time window (on-demand delivery)

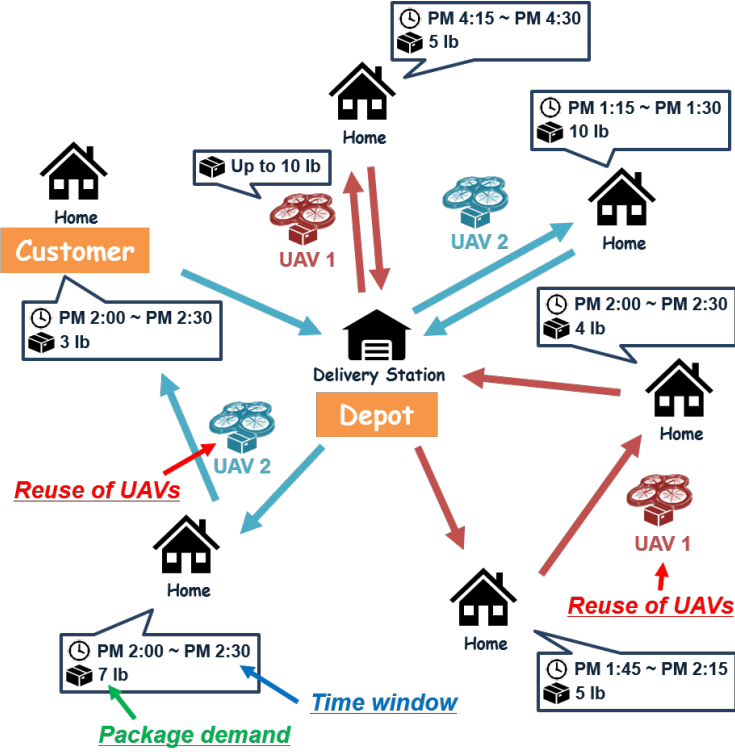


Figure 2.7: Notional example of a Multi-Trip Vehicle Routing Problem with Time Windows (MTVRPTW)

- Reloading packages is conducted at depot only

A notional example of a multi-trip vehicle routing problem with time windows is illustrated in Figure 2.7. This example allows to reuse UAVs if there is time enough to reload packages and recharge/exchange batteries between two routes. This leads to a plan requiring the less number of required UAVs than a plan created by a vehicle routing problem with time windows in Figure 2.6. This is an essential concept of operations for sUAS-based delivery systems due to UAV's short endurance and small payload capacity as discussed in the previous section.

However, a MTVRPTW optimization model does not account for limited endurance. That is, vehicles considered in a MTVRPTW optimization model has mathematically infinite endurance. Except for this limitation, a MTVRPTW optimization model well describes the operations of sUAS-based delivery systems. Thus, this thesis extends a MTVRPTW optimization model by adding the concept of vehicle having limited endurance to address the

operations of sUAS-based delivery systems. The extended optimization model is presented in the following chapter.

2.3 Summary

This chapter discussed fundamental concepts of both the Fixed Point Iteration (FPI) method and the Vehicle Routing Problem (VRP) to provide background enough to address the research questions respectively such that

Research Question 4 (RQ 4) : How can the small VTOL UAV design module and the VRP module be integrated with a FPI method to obtain a converged solution?

Research Question 2 (RQ 2) : What type of a VRP optimization model best represents the operations of sUAS-based delivery systems?

This thesis uses a FPI method to obtain a converged solution of the concurrent UAV design and routing for urban delivery systems previously illustrated in figure 1.21. This chapter showed why the FPI can be utilized to find a converged solution of an MDO problem using mathematical theories.

This thesis also utilizes a VRP optimization to build routes of vehicles for delivery systems which minimize the total cost defined in Equation 1.1. This chapter discussed how a VRP is modeled based on graph theory step by step, and then presented existing VRP optimization models relevant to this thesis. The background knowledge discussed in this chapter would be helpful to capture the novel ideas presented in this thesis in the following chapters.

CHAPTER 3

MATHEMATICAL MODELING OF A SMALL UAS-BASED DELIVERY SYSTEM

This chapter discusses a VRP model capable of addressing operations of sUAS-based delivery systems. The research question associated with this chapter is that

Research Question 2 (RQ 2) : What type of a VRP optimization model best represents the operations of sUAS-based delivery systems?

In general, a VRPTW model is used to model operations of delivery systems for on-demand service. However, to address a VRP optimization model for sUAS-based delivery systems, the VRP model should additionally involve the properties that vehicles can be reused, and they have limited endurance, as stated previously.

Section 3.1 develops a hypothesis for this research question. Section 3.2 presents a new VRP model to address operations of a UAS-based delivery system. Then, Section 3.3 conducts experiments to demonstrate the hypothesis for Research Question 2.

3.1 Development of Hypothesis

While the CVRP and VRPTW formulations, described in Subsection 2.2.6, have successfully modeled operations planning problems for vehicles such as trucks [59, 60, 64], passenger aircraft [65, 66], and cargo aircraft [67, 68], the formulations do not account for the physical and operational properties of sUAS-based delivery systems because these models assume that each vehicle can take a single route during operating hours implying that each vehicle can be used just once during its operating period. However, a sUAS operator may want to reuse individual aircraft throughout the day to make multiple deliveries because sUAVs have limited range and payload capacity. The need to constantly reuse and recharge

sUAVs drives an extension to the CVRP and VRPTW where the recharging policy is explicitly considered in the optimization model. For instance, while refueling a truck takes only a few minutes and be done mid-route, recharging a UAS may take an extended time and can only be done at depot locations (at present). There are two main approaches to describe recharging electric vehicles: the Green Vehicle Routing Problem (GVRP) and MTVRP.

The GVRP was introduced by Erdoğan and Miller-Hooks [69] for alternative fuel-powered vehicle fleets. In the GVRP, each vehicle can travel up to maximum driving range without refueling, and the vehicle can extend its driving range by stopping en route at any refueling stations. In the GVRP model, nodes consists of depots, customers, and refueling stations; this extends more traditional VRPs whose nodes consist of depots and customers. In GVRP models, each recharging station node allows to be visited no more than once. Thus, the GVRP models add a set of dummy nodes for each station to allow each station to be visited multiple times. This model is mathematically similar to the VRP model with satellite facilities [70]. In the VRP model with satellite facilities, a vehicle can replenish fuel and preselected packages in satellite facilities. Schneider et al. [71] introduced the Electric Vehicle Routing Problem with Time Windows (EVRPTW) model, which adds time window constraints into the GVRP model. The EVRPTW model allows the recharging of batteries based on the remaining battery capacity. A continuation of this work [72] extended the EVRPTW with intermediate stops including refueling stations, loading facilities, and combined facilities for both refueling and loading. Goeke and Schneider [73] introduced an EVRPTW model for a mixed fleet of electric and conventional vehicles. Hiermann et al. [74] extended the EVRPTW model by allowing heterogeneous electrical vehicles. However, these GVRP models mainly address larger systems then a last-mile delivery system which is considered in this thesis.

The MTVRP was proposed by Fleischmann [75], which allows a vehicle to stop at the depot to loading additional packages between customer deliveries. This feature can substantially reduce costs by reusing vehicles during operating hours. Because of this

benefit, the MTVRP has recently found significant applications due to the development of electrical vehicles such as green cars and sUAVs [76]. The MTVRP model allows a sUAV to have a journey, defined as multiple round trip flights from a depot, not just a single route or trip, defined as a single round trip flight from a depot. The MTVRP has been modeled by 2-index, 3-index, and 4-index vehicle flow formulations. 2-index models [77] have only arc indices, 3-index models include a vehicle index [78] or trip index [79], and 4-index models [58, 80] contain both the vehicle, trip, and arc indices. Dorling et al. [11] proposed a optimization model for the sUAS delivery problem based on the MTVRP model that optimizes the weight of payloads and batteries concurrently with vehicle routings. The MTVRP has also been extended to account for customer time windows as traditionally modeled by a VRPTW; the MTVRPTW model [81], described in Subsection 2.2.6, has the capability to accommodate the properties MTVRP and VRPTW simultaneously.

The MTVRPTW model approximates a delivery system in that a single depot is used for a fleet of vehicles making deliveries to customers with specific time windows. The MTVRPTW model, also, consider reuse of vehicles by reloading packages at the depot. In this sense, the MTVRPTW model can describes most of characteristics for operations of a sUAS for last-mile delivery system as stated previously. However, the MTVRPTW model considers infinite endurance/range vehicles while sUAVs have short endurance/range. Thus, to address a sUAS-based delivery system, the MTVRPTW model should be extended by adding constraints describing limited endurance or range. For sUAS-based delivery systems using sVTOL UAVs, endurance constraints are preferred because endurance can include effects of take-off and landing for delivery while range cannot address them. These observations inspire a hypothesis for Research Question 2 (RQ 2) such that

Hypothesis for RQ 2 : To address operations of sUAS-based delivery systems, an endurance-constrained MTVRPTW model should be utilized.

The following section will discuss background of the VRP including fundamental concepts to create formulations and existing VRP models relevant to this thesis. Then, Sec-

tion 3.2 will present an endurance-constrained MTRPTW model as an extension of the MTRPTW model, as stated in Subsection 2.2.6, by adding maximum endurance constraints for each sUAV.

3.2 Optimization Model for an Endurance-Constrained MTRPTW ¹

This section presents a mathematical optimization model for sUAS-based delivery as an endurance-constrained MTRPTW model. The model is a 3-index formulation including arc and vehicle indices which can trace a single route. The optimization model is described by a graph, $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, and a fleet of vehicles, $\mathcal{V} = \{1, \dots, m\}$. A graph, \mathcal{G} , contains a set of nodes, $\mathcal{N} = \{0, 1, 2, \dots, n, n+1\}$, and a set of arcs, $\mathcal{A} = \{(i, j) : \forall i, j \in \mathcal{N}, i \neq j, i \neq n+1, j \neq 0\}$.

First, a set of nodes, \mathcal{N} , consists of customers, $\mathcal{C} = \{1, 2, \dots, n\}$, and two depots, $\mathcal{D} = \{0, n+1\}$. In the context of sUAS-based delivery, each customer is regarded as a delivery location. Note that the two depots indicates a single depot, which means \mathcal{D}_0 is only used for departures from the depot while \mathcal{D}_{n+1} represents an arrival at the depot. Let \mathcal{N}_O be the union of the origin depot and customer nodes like $\mathcal{N}_O = \{0, 1, 2, \dots, n\}$ and \mathcal{N}_D be the union of the destination depot and customer node such as $\mathcal{N}_D = \{1, 2, \dots, n, n+1\}$. A customer, \mathcal{C}_i , has a demand for packages, d_i , service time, τ_i , and available time to be visited, $[a_i, b_i]$. In addition, the time for swapping batteries and reloading packages at the depot is notated as τ_{depot} , and the operating hour of the depot is $[a_0, b_0]$. Second, each arc, (i, j) , represents a movement from node i to node j ; it has its own values such as flight distance, $dist_{ij}$, flight time, t_{ij} , and operating cost coefficient, c_{ijk} . Finally, a vehicle has properties such as payload capacity, q , velocity, v , endurance, E , and fixed cost coefficient, c_{fixed} = acquisition cost of a vehicle/# of days in operational life.

In this model, three types of decision variables are defined. The first set of decision variables, x_{ijk} for $\forall (i, j) \in \mathcal{A}$ and $\forall k \in \mathcal{V}$, are allocated to each arc (i, j) with each vehicle

¹This section is submitted for publication [82]

k . Each x_{ijk} is a binary variable and is defined as

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ drives directly from node } i \text{ to node } j \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

The second set of decision variables, s_{ik} for $\forall i \in \mathcal{C}$ and $\forall k \in \mathcal{V}$, indicates the time that a vehicle k starts a service at node i . These variables deal with time as continuous variables.

The last set of decision variables, σ_{ij} for $\forall i, j \in \mathcal{C}$, enables reusability for multiple trips for a vehicle. Each σ_{ij} is a binary variable and is defined as

$$\sigma_{ij} = \begin{cases} 1 & \text{if a vehicle can stop by the depot when the vehicle moves from node } i \text{ to node } j \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$

In most of VRPs, operating time or cost is minimized. Because the proposed model is an endurance-constrained MTRPTW model, the operating time is significantly constrained by time windows for each customer. Therefore, the objective function of the proposed model focuses on minimizing total cost stated in Equation 1.1 such that

$$\text{Min.} \quad c_{fixed} \left(\sum_{k \in \mathcal{V}} \sum_{j \in \mathcal{C}} x_{0jk} - \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} \sigma_{ij} \right) + \sum_{k \in \mathcal{V}} \sum_{i \in \mathcal{N}_O} \sum_{j \in \mathcal{N}_D} c_{ijk} x_{ijk} \quad (3.3)$$

The specific operation requirements are included in the optimization model as a set of constraints of Equation (3.4) ~ (3.17), where K is a large number for the linearization of the constraint, and M is the maximum allowable number of vehicles. First of all, each customer should be visited exactly once by a vehicle, Eq. (3.4). Each vehicle can carry fewer packages than its payload capacity, Eq. (3.5). Moreover, each vehicle should departure from the depot and return to the depot, Eq. (3.6) and (3.8). Eq. (3.9) requires that when a vehicle visits two consecutive nodes i and j , the time at node j should be later than the time at node i . In order for a vehicle to count as reused, the reuse indicator variable, σ_{ij} , must be equal to unity on both the arrival and departure leg to the depot, Eq. (3.10) and Eq.

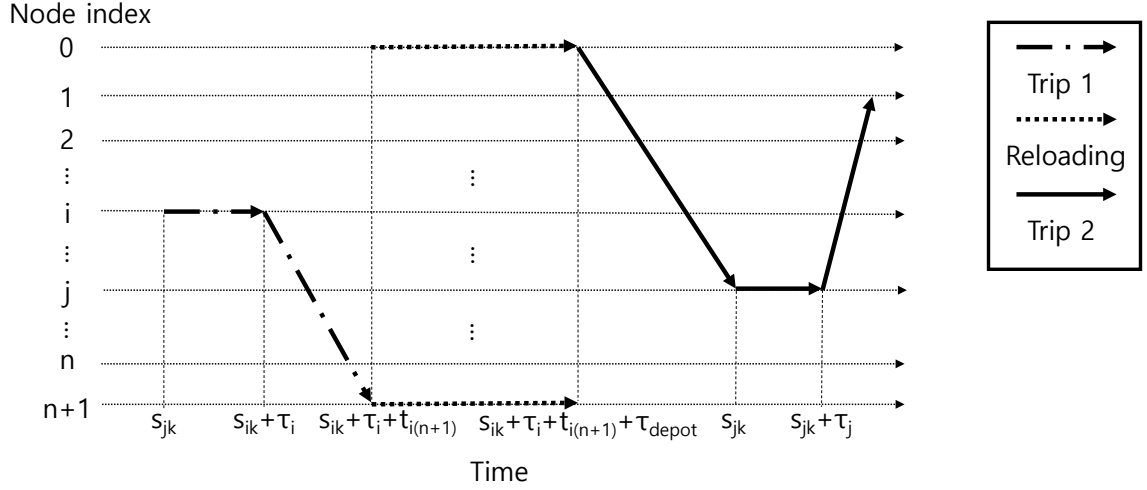


Figure 3.1: Time line for reusability constraint

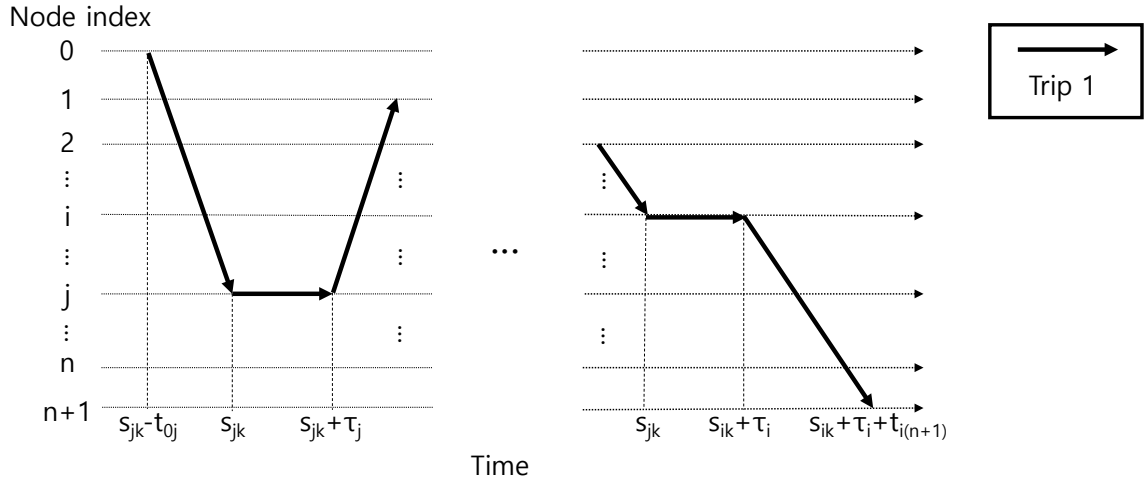


Figure 3.2: Time line for endurance constraint

(3.11) respectively. The number of vehicles which operate concurrently should be less than the maximum allowable number of vehicles, M , Eq. (3.12). When a vehicle can be reused, the time for swapping batteries and reloading packages at the depot should be guaranteed, Eq. (3.13), as illustrated in Fig. 3.1. On the other hand, each vehicle should operate within its endurance for a route or trip, Eq. (3.14), as shown in Fig. 3.2. Finally, Eq. (3.15) are bounding constraints for s_{ik} , and Eq. (3.16) and (3.17) are integrity constraints for x_{ijk} and σ_{ij} .

$$\begin{aligned}
s.t. \quad & \sum_{k \in \mathcal{V}} \sum_{j \in \mathcal{N}_D} x_{ijk} = 1 & \forall i \in \mathcal{C} & (3.4) \\
& \sum_{i \in \mathcal{C}} d_i \sum_{j \in \mathcal{N}_D} x_{ijk} \leq q & \forall k \in \mathcal{V} & (3.5) \\
& \sum_{j \in \mathcal{N}_D} x_{0jk} = 1 & \forall k \in \mathcal{V} & (3.6) \\
& \sum_{i \in \mathcal{N}_O} x_{ihk} - \sum_{j \in \mathcal{N}_D} x_{hjk} = 0 & \forall h \in \mathcal{C}, \forall k \in \mathcal{V} & (3.7) \\
& \sum_{i \in \mathcal{N}_O} x_{i,n+1,k} = 1 & \forall k \in \mathcal{V} & (3.8) \\
& s_{ik} + t_{ij} + \tau_i - s_{jk} \leq K(1 - x_{ijk}) & \forall i \in \mathcal{N}_O, \forall j \in \mathcal{N}_D, \forall k \in \mathcal{V} & (3.9) \\
& \sum_{j \in \mathcal{C}} \sigma_{ij} - \sum_{k \in \mathcal{V}} x_{i(n+1)k} \leq 0 & \forall i \in \mathcal{C} & (3.10) \\
& \sum_{i \in \mathcal{C}} \sigma_{ij} - \sum_{k \in \mathcal{V}} x_{0jk} \leq 0 & \forall j \in \mathcal{C} & (3.11) \\
& \sum_{k \in \mathcal{V}} \sum_{j \in \mathcal{C}} x_{0jk} - \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} \sigma_{ij} \leq M & & (3.12) \\
& s_{ik} + \tau_i + t_{i(n+1)} + \tau_{depot} + t_{0j} - s_{jk} \leq K(1 - \sigma_{ij}) & \forall i, j \in \mathcal{C}, \forall k \in \mathcal{V} & (3.13) \\
& s_{ik} + \tau_i + t_{i(n+1)} - (s_{jk} - t_{0j}) - E \leq K(2 - x_{i(n+1)k} - x_{0jk}) & \forall i, j \in \mathcal{C}, \forall k \in \mathcal{V} & (3.14) \\
& a_i \leq s_{ik} \leq b_i & \forall i \in \mathcal{C}, \forall k \in \mathcal{V} & (3.15) \\
& x_{ijk} \in \{0, 1\} & \forall i \in \mathcal{N}_O, \forall j \in \mathcal{N}_D, \forall k \in \mathcal{V} & (3.16) \\
& \sigma_{ij} \in \{0, 1\} & \forall i, j \in \mathcal{C} & (3.17)
\end{aligned}$$

3.3 Experiments

This section designs and executes two experiments to demonstrate the effects of both reusability and maximum endurance constraints on a solution schedule such that

Experiment 1 : To find what type of VRP model best represents a UAS package delivery network, this experiment compares three models: VRPTW, endurance-constrained VRPTW, and endurance-constrained MTVRPTW models.

Experiment 2 : To address a tendency of solutions by reusability and maximum endurance conditions, this experiment executes sensitivity analysis of the constraints with two models: VRPTW and endurance-constrained MTVRPTW models.

Specifically, Experiment 1 executes simulations with three VRP optimization models: a VRPTW model, as described in Section 2.2.6, is the baseline. An endurance-constrained VRPTW model has all constraints of the baseline and maximum endurance constraints additionally. An endurance-constrained MTVRPTW model has all constraints of the MTVRPTW model and reusability constraints. Experiment 2 demonstrates a tendency of solution by sensitivity analysis. When the tendency is reasonable, the endurance-constrained MTVRPTW model can address its necessity to describe operations of a sUAS-based delivery system.

For two experiments, a benchmarking data set² for VRPTW models, called R101.25, is used as an input graph; because its total cost optimized solution is already known in terms of the minimum total distance, the effects of additional constraints on a solution can be shown clearly. The graph includes 25 customers having up to 29 package demand with specific time windows. Every service time for delivery is 10 unit time. For this data set, a common assumption is that homogeneous vehicles having 200 payload capacity deliver packages at unit speed.

²<http://w.cba.neu.edu/~msolomon/problems.htm> (accessed 27 January 2019)

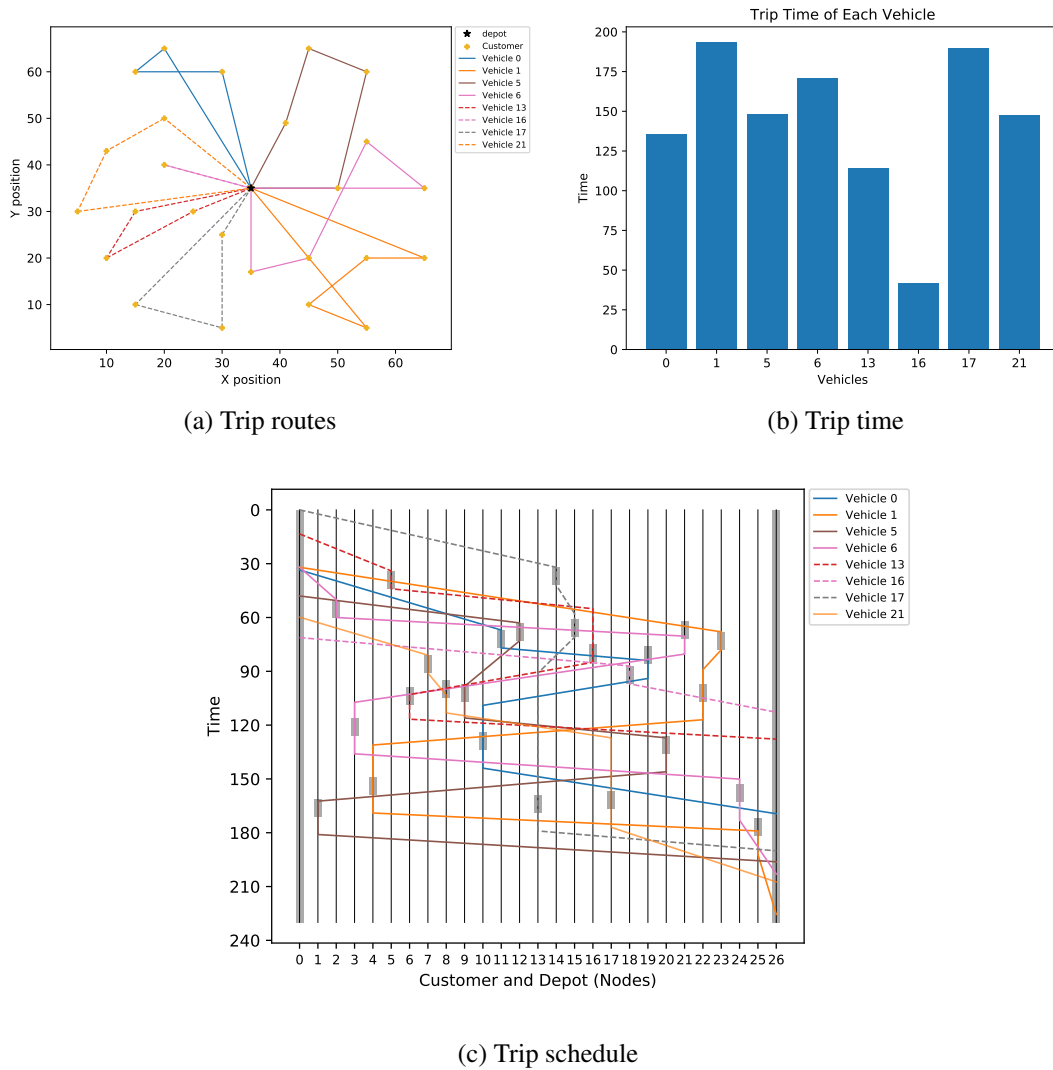


Figure 3.3: Total cost optimized solution of a VRPTW model (baseline)

3.3.1 Experiment 1: Comparison of Optimization Models

Using the input graph, three simulations for VRPTW, endurance-constrained VRPTW, and endurance-constrained MTVRPTW models are conducted. First, the solution schedule of the baseline, VRPTW model, requires the total unit distance of 617.1 and 8 vehicles that are well known solution. The schedule is shown in Figure 3.3. Each route of vehicles is illustrated in Figure 3.3a. The maximum trip time is 193.5 unit time required by the vehicle 1 in Figure 3.3b. The total cost optimized schedule is described in a time-space

plot like Figure 3.3c; each route starts at node 0 and ends at node 26, and gray-colored boxes represent time windows at each node. This is the baseline result.

Second, to address short endurance vehicles, the maximum endurance constraints in Equation (3.14) are added to the baseline model, which is an endurance-constrained VRPTW. Note that the endurance-constrained VRPTW model has maximum endurance constraints, but it does not have reusability constraints. In this simulation, the maximum endurance is assumed as 100 unit time which is about a half of the maximum trip time of the baseline result. Then, the simulation is executed with the same input graph. The solution schedule of this model is shown in Figure 3.4. This solution requires the total unit distance of 793.6 and 13 vehicles. Figure 3.4b shows that maximum endurance of each vehicle is restricted up to 100 unit time. This is the purpose of the maximum endurance constraints. The result demonstrates that the endurance-constrained VRPTW model can address short-endurance vehicles by maximum endurance constraints.

Lastly, to deal with reuse of vehicles, the reusability constraints in Equation (3.10) ~ (3.13) are added to the endurance-constrained VRPTW model. This leads to the mathematical optimization model of the endurance-constrained MTVRPTW presented in Section 3.2, and its simulation result is visualized in Figure 3.5. The solution schedule requires 10 vehicles with the total distance of 808.2, while the longest trip time is constrained up to 100 unit time due to the maximum endurance constraints. The main difference between this model and others is that a vehicle can have multiple routes; some vehicles such as vehicle 3 and 9 take multiple trips in Figure 3.5a and these vehicles return to node 0 for reloading packages as shown in Figure 3.5b. This result shows that the endurance-constrained MTVRPTW model allows vehicles to be reused. This is the purpose of the reusability constraints.

To address operations of a sUAS-based delivery system, a new VRP optimization model, as an extension of the VRPTW model, should additionally consider reuse of vehicles and limited endurance as mentioned previously. The result shows only the endurance-constrained MTVRPTW model can create a solution involving the effects of reuse of ve-

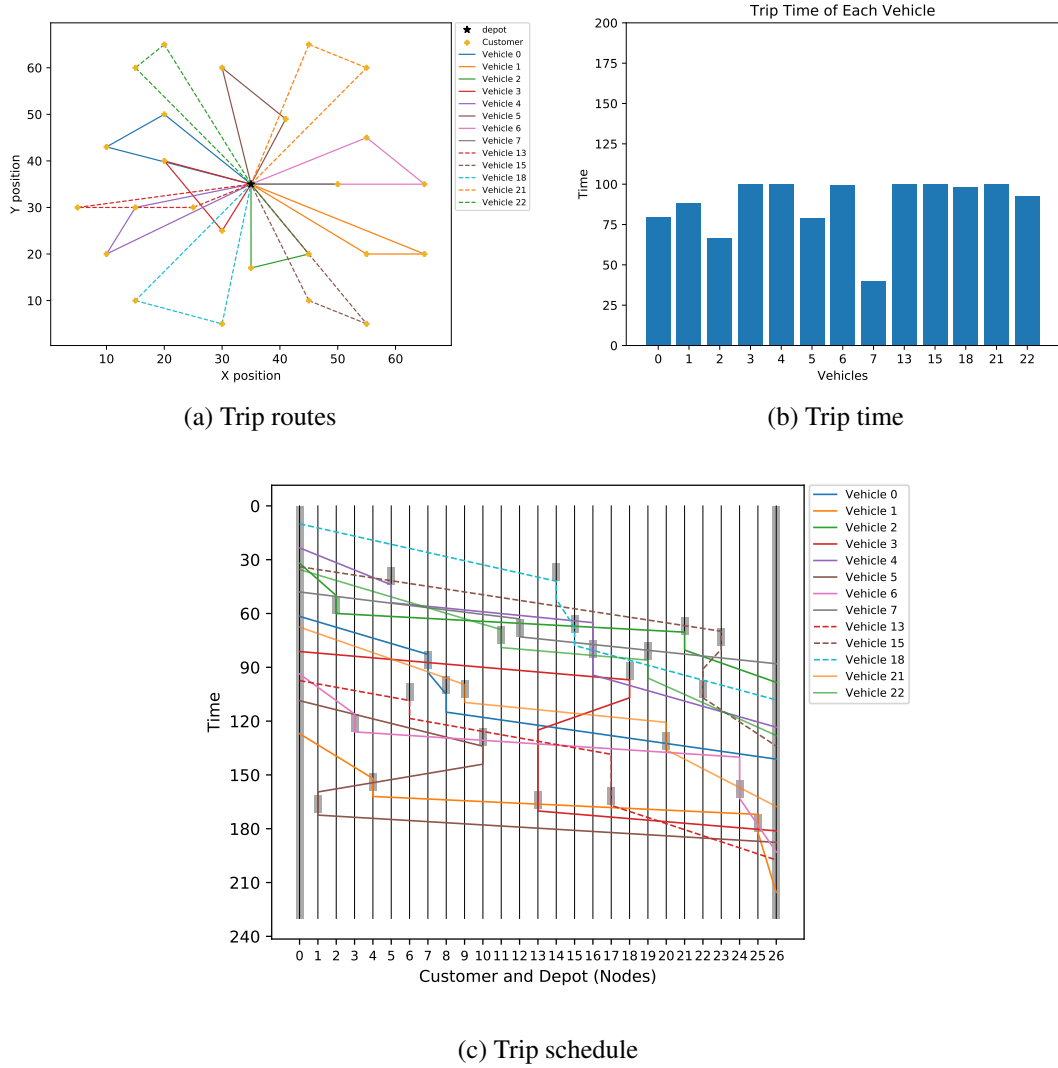
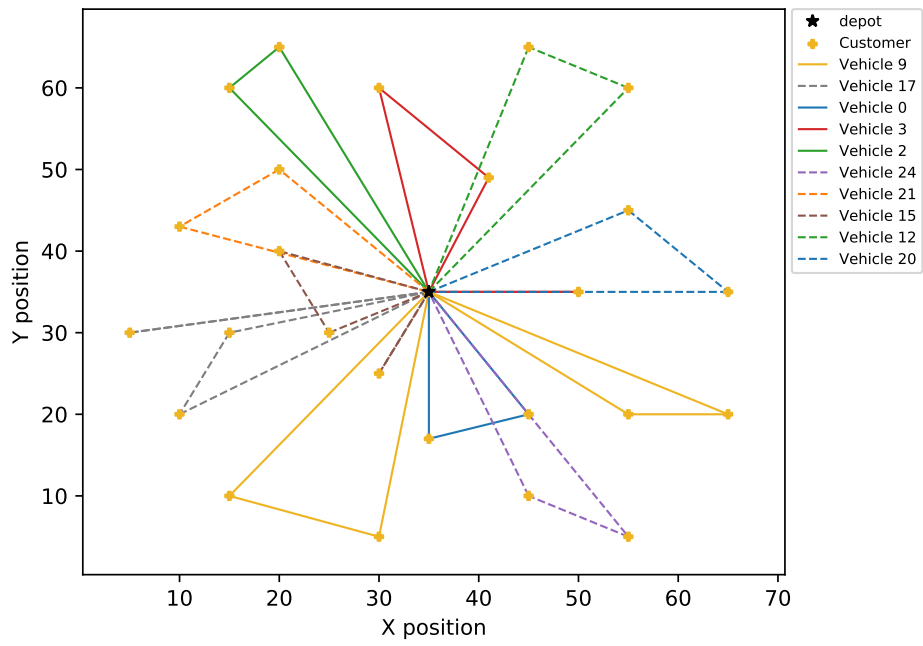


Figure 3.4: Total cost optimized solution of an endurance-constrained VRPTW model

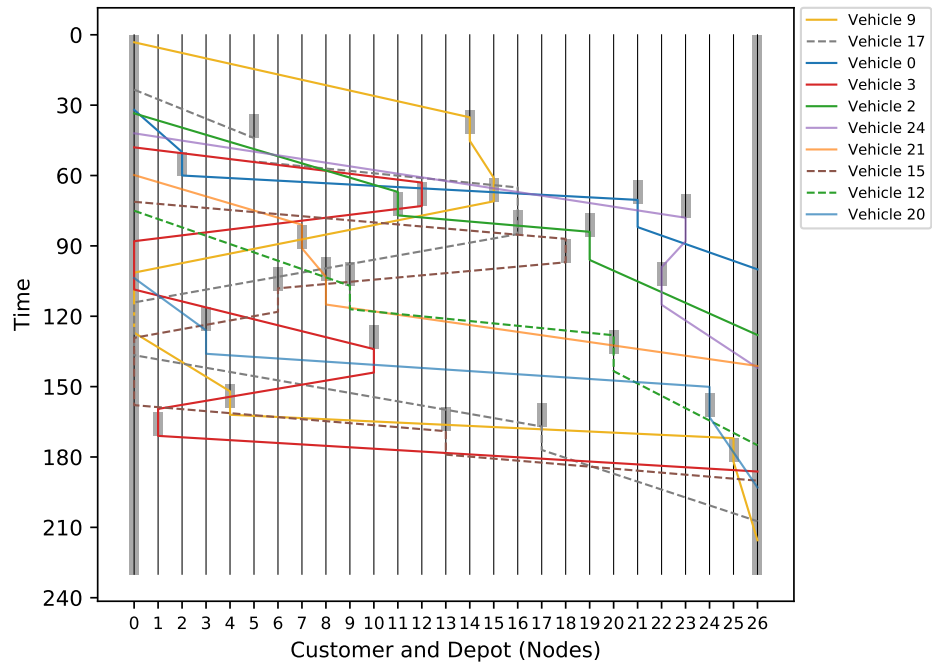
hicles and limited endurance. Therefore, the endurance-constrained MTVRPTW model should be used to describe operations of a sUAS-based delivery system.

3.3.2 Experiment 2: Sensitivity Analysis

Experiment 2 executes sensitivity analysis of maximum endurance and reusability constraints. To do that, 16 scenarios are created by various maximum endurance and reusability options. Note that in this experiment, fixed cost coefficient, c_{fixed} , is assumed as 100 unit cost to disincentivize the use of excessive vehicles when calculating the objective func-

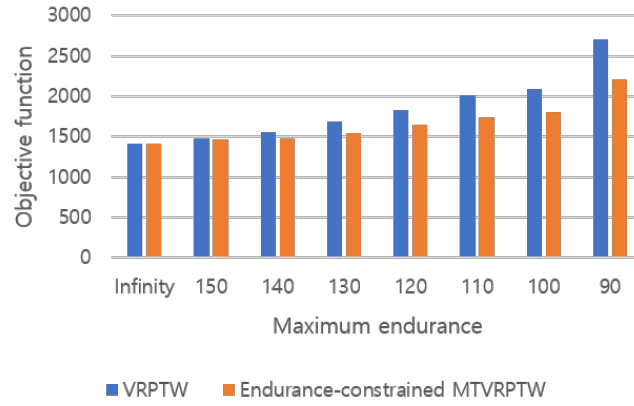


(a) Journey routes

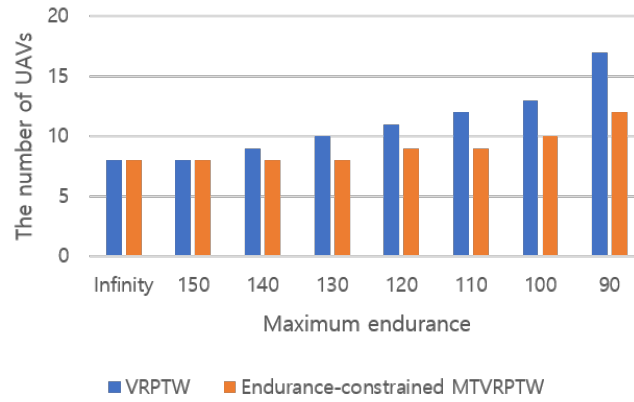


(b) Journey schedule

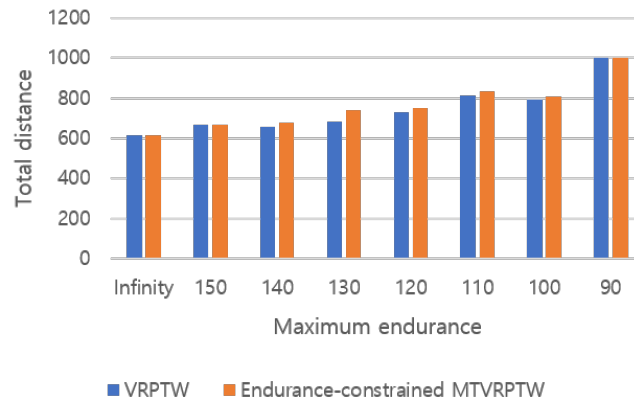
Figure 3.5: Total cost optimized solution of an endurance-constrained MTVRPTW model



(a) Effects of endurance on objective function values of both the VRPTW and the endurance-constrained MTVRPTW models



(b) Effects of endurance on the number of required vehicles of both the VRPTW and the endurance-constrained MTVRPTW models



(c) Effects of endurance on total distance of both the VRPTW and the endurance-constrained MTVRPTW models

Figure 3.6: Summary of simulation results

Table 3.1: Summary of total cost optimized schedules of the 16 scenarios

No.	Reusability	Endurance	Obj. func.	# of UAVs	Total dist.	Max. trip time
1	No	Infinite	1417.1	8	617.1	193.5
2		150	1470.7	8	670.7	148.6
3		140	1559.2	9	659.2	138.1
4		130	1686.5	10	686.5	130
5		120	1831.3	11	731.3	120
6		110	2013.8	12	813.8	110
7		100	2093.6	13	793.6	100
8		90	2701.6	17	1001.6	90
9	Yes	Infinite	1417.1	8	617.1	193.5
10		150	1467.9	8	667.9	150
11		140	1478.8	8	678.8	140
12		130	1540.3	8	740.3	130
13		120	1650.9	9	750.9	120
14		110	1734	9	834	110
15		100	1808.2	10	808.2	100
16		90	2201.6	12	1001.6	90

tion in Equation 3.3. However, in a framework for concurrent UAV design and routing for urban delivery systems presented in this thesis, the fixed cost coefficient is fed by an aircraft sizing and synthesis module.

The results are visualized in Figure 3.6 in terms of objection function, the number of vehicles, and the total distance respectively and summarized in Table 3.1. The first case is the result of the baseline. Next, the results of cases from 2 to 8 describe total cost optimized schedules of the endurance-constrained VRPTW model. The other cases show results of the endurance-constrained MTVRPTW model. First, as maximum endurance of vehicles is decreased, the objective function increases. However, the objective function of the VRPTW model increases more sharply than the MTVRPTW. This is expected because by reusing vehicles, the MTVRPTW model reduces the fixed cost of the delivery system as shown in Figure 3.6b. Second, as maximum endurance is decreased, the total distance increases as

illustrated in Figure 3.6c. In this result, the value of the VRPTW model is the lower bound of the value of the MTVRPTW model. However, since the effects of the total distance on the objective function is less than the number of vehicles, the objective function increase as the maximum endurance decreases. These results also show that why an endurance-constrained MTVRPTW model needs to be for sUAS-based delivery systems.

3.4 Conclusions

This chapter started from the research question such that

Research Question 2 (RQ 2) : What type of a VRP optimization model best represents the operations of sUAS-based delivery systems?

To address this research question, a literature review was conducted, and observed that the VRPTW optimization model describes a on-demand delivery system and the MTVRPTW optimization model represents operations of a delivery system including the vehicle reuse concept. However, the MTVRPTW optimization model cannot address the property of vehicles of limited endurance. This observation inspired the hypothesis such that

Hypothesis for RQ 2 : To address operations of sUAS-based delivery systems, an endurance-constrained MTVRPTW model should be utilized.

To prove the hypothesis, experiments were executed. Experiment 1 demonstrated that only the endurance-constrained MTVRPTW model can address the characteristics of both reuse of vehicles and limited-endurance vehicles. Furthermore, Experiment 2 also confirmed that the endurance-constrained MTVRPTW model should be utilized for a delivery system using short endurance vehicles to obtain a total cost optimized schedule which minimizes total cost. This result leads to a conclusion that Hypothesis for RQ 2 is substantiated.

CHAPTER 4

CONSTRUCTION OF A TWO-LAYERED URBAN FLIGHT NETWORK ¹

The previous chapter presented an endurance-constrained MTRPTW optimization model. This chapter discusses a way how to create an input graph of a VRP model to address the research question such that

Research Question 3 (RQ 3) : What type of an input graph of a VRP model best represents collision-free paths in the urban environment for sUAS-based delivery systems?

To deal with the urban environment as an input graph of a VRP, the graph should provide collision-free paths in terms of urban obstacles such as tall buildings and restricted airspace, as described previously. This chapter presents a framework to create an input graph for the urban area of San Diego, CA. San Diego is selected as a mission area of this thesis due to the fact that

- Point cloud data scanned by airborne Light Detection and Ranging (LiDAR) sensors, which have a good resolution, is fully obtainable.
- In a relatively small area, there are a number of urban obstacles and restricted airspace. This allows to conduct computationally efficient simulations for various environment conditions using relatively small-size point cloud data.

Note that the San Diego model is just used to make the framework for creating an urban flight network. The developed framework in this chapter requires both point cloud data of an urban area and a set of polyhedrons describing restricted airspace for the urban area as its input. Because the framework creates collision-free paths based only on the input, if both point cloud data of an urban area and a set of polyhedrons describing restricted

¹This chapter is submitted for publication [82]

airspace for the urban area are available, the developed framework can be utilized to build collision-free paths for the urban area.

Section 4.1 develops a hypothesis for Research Question 3. Section 4.2 presents a framework to create a two-layered urban flight network with the San Diego example. Then, Section 4.3 conducts experiments to demonstrate the hypothesis with the framework for creating a flight network in the urban environment.

4.1 Development of Hypothesis

Because point-to-point paths cannot be used in an urban environment, as stated previously, collision-free paths need to be used. Path planning methods to create collision-free paths can be categorized into five groups [83]: stochastic methods, road map methods, potential field approaches, geometric methods, and control-theory-based methods with grids. Stochastic methods, such as a Rapidly-exploring Random Tree (RRT) method [84], gradually expand paths by using randomly-selected points from an initial point to a target point. Road map methods model empty space as a network consisting of piecewise linear paths, and then they create paths based on the network [85]. Potential field methods construct a force map; waypoints create attractive forces to pull vehicles while obstacles generate repulsive forces to push vehicles [85]. Geometric methods describe vehicles and obstacles as objects in a space and then check whether there is any collision or not using geometric boundaries of vehicles [86]. Control-theory-based methods also have successfully solved collision-free path-planning problems. For instance, collision cone approaches [87, 88] guarantee that a vehicle moves to a waypoint and avoids moving obstacles. Approaches based on Mixed-Integer Linear Programming (MILP) [89, 90] use binary constraints to determine whether collisions happen or not based on obstacles modeled in design space.

The collision-free path-planning methods in the five groups have utilized for various missions. However, to address collision-free paths for sUAS-based delivery systems, road map methods are considered in this thesis due to the fact that

- Stochastic methods are created randomly. This leads an input graph of a VRP model to have different attribute values for edges whenever the graph is created. This characteristic as an input graph of a VRP model is not preferred because the VRP is a deterministic problem. Moreover, if there is no possible path, the methods fail to stop.
- Potential field methods create a path at once. However, to build an input graph of a VRP model, a myriad of paths are required to describes all possible options. Furthermore, the number of options is increased exponentially as the number of nodes of a graph increases. Thus, for a large-scale input graph, potential field methods could require too long time to create the graph. Moreover, for the complicated environment, vehicles could be trapped in a local minimum point of the force map [85].
- Geometric methods and control-theory-based methods requires dynamic inputs because they are on-line methods. However, to create dynamic inputs, a VRP model should be solved. Thus, in the step of building an input graph of a VRP, the on-line methods cannot be used.

Road map methods are independent from the size of an input graph of a VRP model, and they are dependent on the obstacles existing in a mission area. For multi-vehicle missions, if a road map is created once, then it can be reused to create paths for all vehicles [91]. Therefore, this thesis considers a road map method to create an input graph of the endurance-constrained MTRPTW optimization model presented previously. Table 4.1 shows the comparison of the five groups of path-planning methods.

Road map methods can be further broken down into two categories [85]: cell decomposition method, navigation mesh method, and graph-based method. First, the cell decomposition method divides the non-obstructed area of the area of interest (AOI) into regular or irregular grids. This method requires substantial computational resources for adequate accuracy due to the need for highly refined grids. Next, the navigation mesh method, also called the meadow map [92], decomposes empty space into convex polygonal grids, which

Table 4.1: Comparison of the five groups of path-planning methods

Group	Characteristics	Advantage	Disadvantage
Stochastic methods	Extends a path randomly	One of the fastest algorithms for path planning	Varies whenever a graph is created and fails to stop when there is no possible path
Potential field methods	Constructs a force map	A simple and fast algorithm for real-time applications	Could be trapped in a local minimum
Geometric methods	Describes vehicles and obstacles as objects	A simple and fast algorithm for real-time applications	An on-line method requiring dynamic inputs
Control-theory-based methods	Includes vehicle dynamics	Creates precise paths	An on-line method requiring dynamic inputs
Road map methods	Models empty space as a network	Provides the safest paths	Could not address the shortest paths

creates fewer grid cells than cell decomposition methods. Last, the graph-based method creates a graph based on a set of piecewise linear collision-free paths directly, not based on empty space decomposition; this method also avoids the computational expense required for cell decomposition methods to create paths. Graph-based methods can be further decomposed into visibility graphs and Voronoi diagrams.

The visibility graph solves Euclidean shortest path (ESP) problems; this graph is created by modeling each corner point of obstacles as a node and then connecting all mutually visible nodes [93, 94]. Because each arc of the visibility graph guarantees a collision-free path, paths created from the visibility graph avoid obstacles. For more practical problems, the visibility graph can be generalized with growing obstacles modeled by an obstacle itself and vehicle's shape and size [93]. This approach guarantees a collision-free path for a vehicle that is described by a polygon or a circle rather than a point. However, the required computational power to create a visibility graph increases rapidly when dealing with nu-

merous obstacles. To address this limitation, the visibility-graph-based ESP methods have been improved to reduce required computational resources [95, 96, 97].

A Voronoi diagram is a geometric structure which divides a plane with given set of points, called sites, in keeping with the nearest-neighbor rule: every point in the plane is associated with a site closest to it [98]. The points associated with each site form a subplane called a Voronoi cell, and its boundaries are called Voronoi edges. In the context of path planning, a Voronoi diagram forces each point on a Voronoi edge to maximize the distance to the closest sites [99]. Thus, if a scenario of obstacles is modeled by sites or polygonal sets, a Voronoi diagram builds Voronoi edges that represent collision-free paths.

The Voronoi diagram has been successfully utilized for path-planning problems for robots [100, 101], ships [102], and UAVs [103, 104]. To address UAV path planning problems, a mountainous terrain environment [105], an urban environment [91], and restricted airspace [106, 107] have been modeled as a Voronoi diagram.

From this literature review, there are two observations: the first observation is that a Voronoi diagram can be more efficiently created with a large number of urban obstacles than a visibility graph. The second observation is that a path based on the visibility graph tends to stay as close as possible to obstacles, but sUAV's routes would be better to keep away from obstacles as much as a safety distance. A Voronoi diagram is created based on the bisection of adjacent obstacles, which means it provides the safest route that passes among obstacles. Thus, the Voronoi diagram has the capability of creating a collision-free path with an adequate distance from obstacles. These observations inspire the hypothesis such that

Hypothesis for RQ 3 : To address collision-free paths as an input graph of a VRP for the urban environment, a Voronoi-diagram-based method should be utilized.

The following section presents a framework for creating a two-dimensional input graph for a VRP model based on a Voronoi diagram to guarantee collision-free paths.

4.2 Creating a Framework for a Two-Layered Urban Flight Network

The size of a Voronoi diagram depends on the number and shape of obstacles in the given environment. If a Voronoi diagram is directly transferred as an input graph of a VRP model, the size of the graph increases because the Voronoi diagram includes intermediate points to describe collision-free paths. However, a VRP model needs information related to a depot and delivery locations only. Because a VRP is an NP-hard problem, the required computation power to solve the problem is exponentially increased as the problem size of a VRP increases.

In order to obtain a smaller and more computationally efficient VRP, this thesis presents a framework to create a two-layered urban flight network described in Figure 4.1. The mission area in this thesis is shown in Figure 4.2, which is San Diego, CA, as stated previously. First, from point cloud data, an urban model is created. Second, urban obstacles such as tall buildings and restricted airspace are identified as a set of polygons in the mission area. Third, to address collision-free paths, a Voronoi diagram is created as a low-level flight network— a graph representing physical collision-free paths. Lastly, information which is only directly relevant to a VRP is extracted from the Voronoi diagram; this high-level network is an abstraction of the Voronoi diagram and contains arcs which travel between customers and depots such that arcs represent the collision-free path information in the Voronoi diagram.

Through this two-layered network approach, an input graph of a VRP is reduced to produce a smaller problem size. Moreover, the low-level urban flight network has no effect on the problem size of a VRP, and more detailed Voronoi diagrams can be utilized with similar performance of solving a traditional VRP. The following subsections describe each step of this framework in more detail with intermediate results for each step in Figure 4.1.

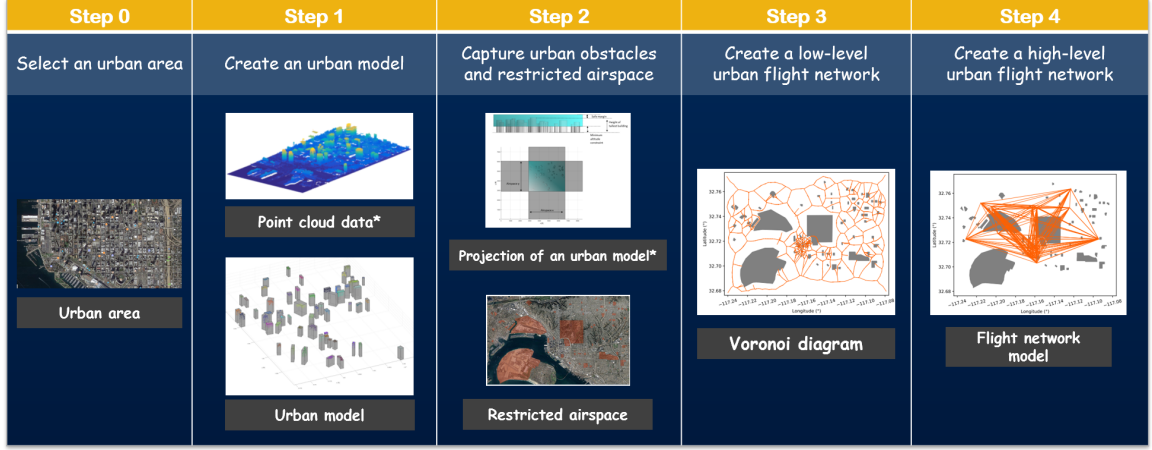


Figure 4.1: The framework for building a two-layered urban flight network

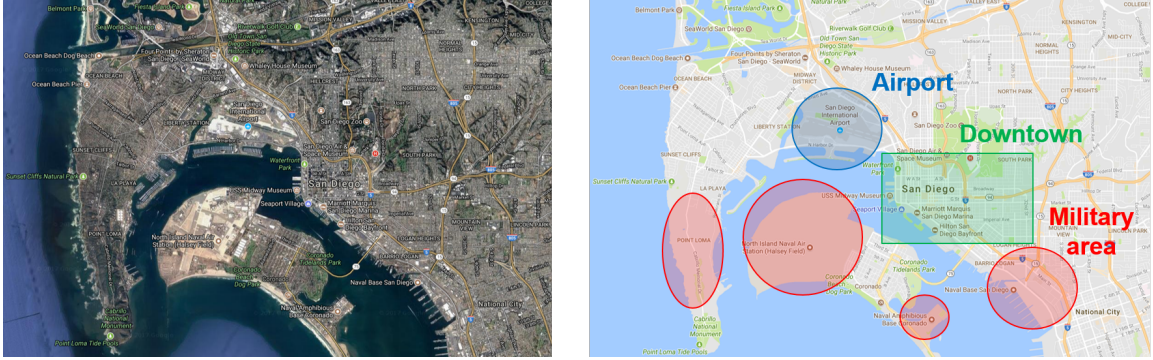


Figure 4.2: Mission area: San Diego, CA (Google Image)

4.2.1 Step 1: Creating an Urban Model

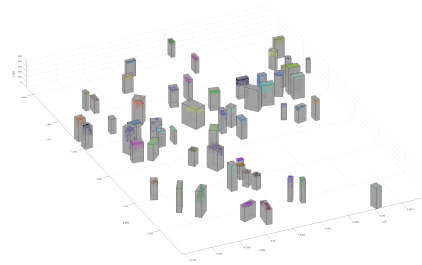
As the first step of the framework to create a two-layered urban flight network in Figure 4.1, an urban topographic model that includes the mission area needs to be obtained. This model should include the configuration and location of each urban obstacle sUAVs have to avoid. In this paper, point cloud data of San Diego, CA collected by an airborne LiDAR sensor is utilized to create the urban model. To extract urban obstacles from the point cloud, this thesis uses the rapid, data-driven, and grid-based urban modeling method proposed by Choi [108]. This method provides a set of three-dimensional corner points describing a urban obstacle in the given area such as a tall building.

4.2.2 Step 2: Capturing Urban Obstacles and Restricted Airspace

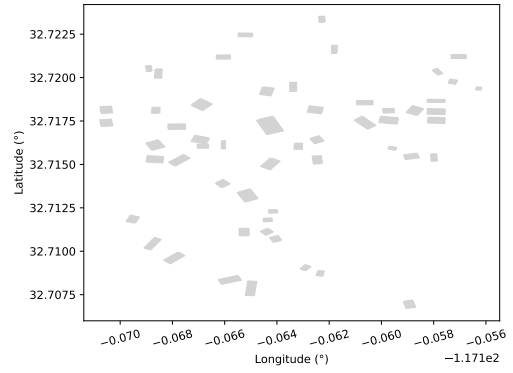
The second step of the framework to create a two-layered urban flight network in Figure 4.1 is capturing urban obstacles and restricted airspace because to build a feasible trajectory, urban obstacles and regulatory requirements must be accounted for. Two notional requirements are considered in this step: a maximum operating altitude of 400 feet above ground level and predefined airspace restrictions for sUAVs such as airports, military facilities, national parks, and schools.

The first step in modeling airspace restrictions is to utilize the point-cloud data used for urban modeling to create specific obstacles for sUAV paths. The desired Above Ground Level (AGL) operating altitude can be modeled as a surface above an urban area's topography. The intersection between the surface and the urban model becomes a set of obstacles which are described as a set of polygons. The effect of operating altitude on identifying urban obstacles of San Diego, CA is illustrated in Figure 4.3. The case with 250 feet operating altitude, which is shown in Figure 4.3b, captures 55 urban obstacles, and 300 feet and 350 feet cases, which are illustrated in Figure 4.3c and Figure 4.3d, seize 34 and 25 urban obstacles respectively. This implies that operating altitude is a key factor which determines the complexity of an air map. Note that this thesis only considers paths which go around obstacles, not over them because in most cases, a path going around is a shorter path, and this assumption makes a path-planning problem considered in this thesis a two-dimensional problem.

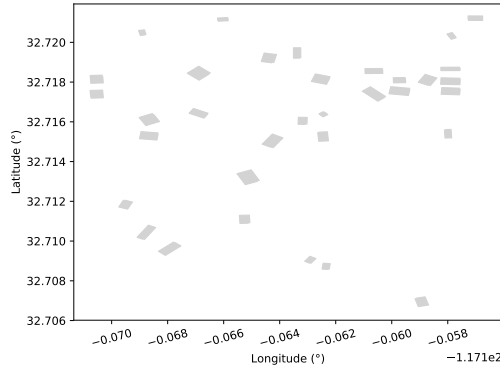
While urban obstacles can be extracted from a point cloud of urban topography, restricted airspace is defined by regulations and usage of land. To account for restricted airspace, each segment of restricted airspace can be modeled as an urban obstacle. Once the two types of urban obstacles are identified, the operating space can be modeled as a surface with a set of obstacles described by polygons as shown in Figure 4.4. Figure 4.4b is the operating space with urban obstacles in Figure 4.3b and restricted airspace in Figure 4.4a.



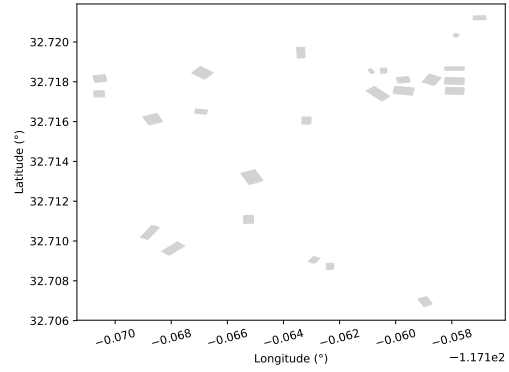
(a) Urban obstacles



(b) Operating space with 250 feet operating altitude



(c) Operating space with 300 feet operating altitude

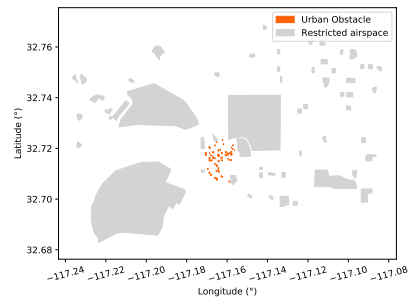


(d) Operating space with 350 feet operating altitude

Figure 4.3: The effect of operating altitude on capturing urban obstacles



(a) Restricted airspace



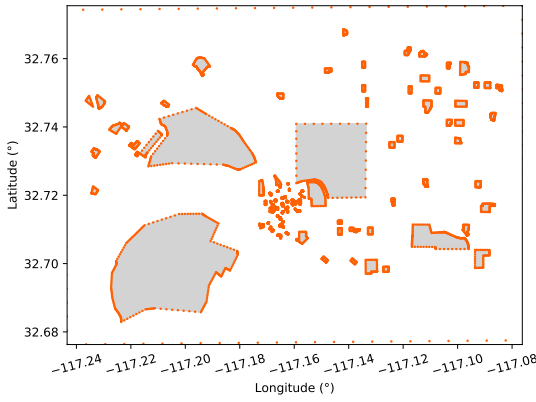
(b) Operating space with urban obstacles and restricted airspace

Figure 4.4: Restricted airspace and operating space with polygonal obstacles

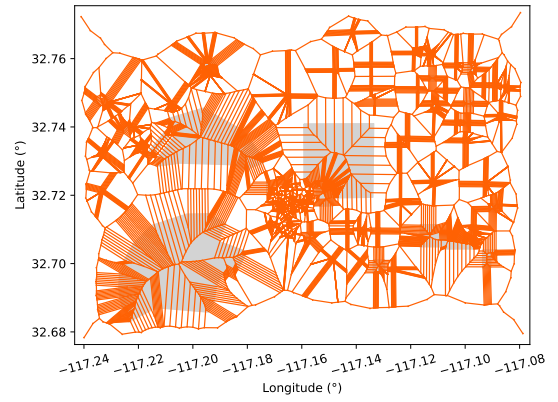
4.2.3 Step 3: Constructing a Low-level Urban Flight Network

In third step of the framework to create a two-layered urban flight network in Figure 4.1, a low-level urban flight network is established as a road map by using a Voronoi diagram to address collision-free paths. The procedure of creating a Voronoi diagram is illustrated in Figure 4.5. The first step is obtaining sample points, commonly referred to as sites, which describe outlines of urban obstacles, which is illustrated in Figure 4.5a. Because a Voronoi diagram is created based on sites, more sites can capture more detailed collision-free paths. The second step is creating a Voronoi diagram which consists of Voronoi vertices and edges, which is visualized in Figure 4.5b. There are two types of Voronoi edges: bounded and unbounded. A bounded edge is defined by two Voronoi vertices while an unbounded edge is made from a single Voronoi vertex. The last step shown in Figure 4.5c is filtering out extraneous Voronoi vertices and edges; for instance, vertices within an obstacle, unbounded edges, and edges that intersect with a part of an obstacle can be removed. Another critical filter that must be applied to the Voronoi diagram is the removal of edges which cannot physically accommodate a sUAV's path— for instance an edge could exist between two obstacles but the physical clearance between the obstacles would not allow for safe sUAV flight. The clearance could be adjusted based on the dimensions of the sUAV being considered. Thus, Voronoi vertices and edges which provide less than 20 feet of clearance, which is assumed as a purpose of developing this framework, are removed. After the filtering process, there are still redundant edges. However, because the redundant edges could be used to connect a Voronoi diagram with nodes of a graph of a VRP optimization model, they are kept. Although they still exist, a path-planning method does not select the redundant edges because if the edges are selected, the path is trapped.

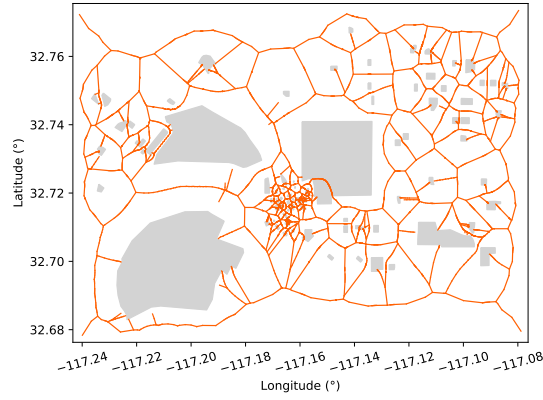
In addition, when a Voronoi diagram is used in a path planning field, an artificial boundary needs to be added to the diagram as a set of sites; without the boundary, a Voronoi diagram does not account for detours around the outside obstacles as shown in Figure 4.6. Thus, this thesis add an artificial boundary for a given mission area to address the detours.



(a) Step 1: Sampling points from obstacles

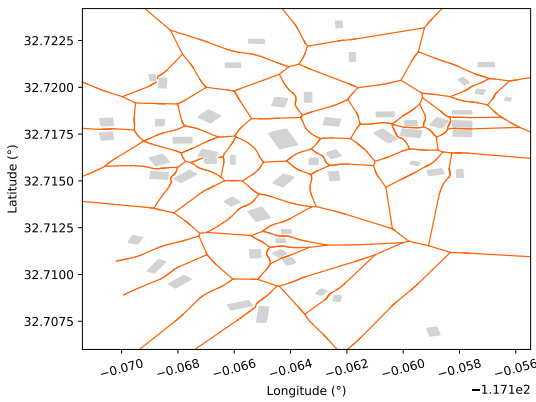


(b) Step 2: Creating Voronoi vertices and edges

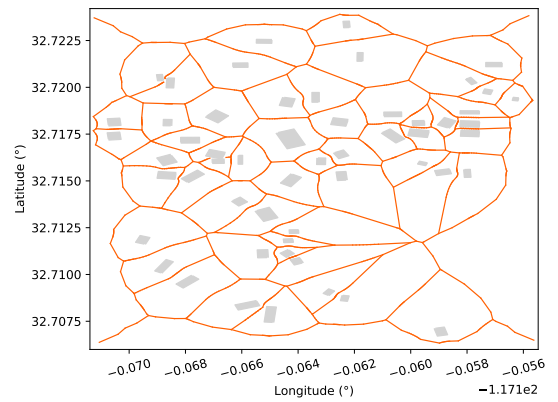


(c) Step 3: Filtering Voronoi vertices and edges

Figure 4.5: Procedure of creating a low-level urban flight network



(a) Road map created without an artificial boundary



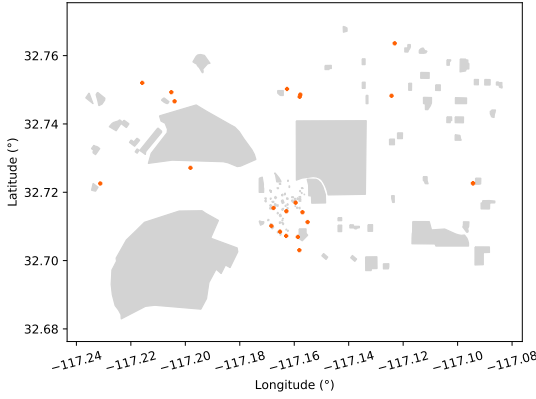
(b) Road map created with an artificial boundary

Figure 4.6: Effect of an artificial boundary on a road map

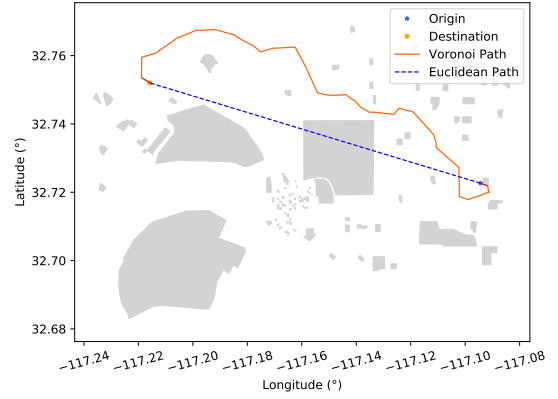
4.2.4 Step 4: Building a High-level Urban Flight Network

The last step of the framework to create a two-layered urban flight network in Figure 4.1 is building a high-level urban flight network. A high-level urban flight network is built by extracting information directly relevant to a VRP from the low-level urban flight network as shown in Figure 4.5c. The relevant information is created from Voronoi paths between customers and depots. There are two steps to create Voronoi paths. First step is representing each node of the VRP in the low-level urban flight network. Each node is either a depot or a delivery location as shown in Figure 4.7a. For simplicity, each node can be connected to the closest Voronoi vertex using an additional arc. The second step is finding the shortest paths between customers and depots; any path finding algorithms from graph theory can be utilized. In this paper, Dijkstra's shortest path algorithm is adopted to find a Voronoi path and to obtain its network distance from the road map created by a Voronoi diagram. Figure 4.7b and 4.7c show comparisons between Euclidean paths and Voronoi paths. In the first example, a Voronoi path is established using a detour created by an artificial boundary. In the second example, when vehicle passes through downtown San Diego, the Voronoi path is built by a combination of safe edges which avoid from objects by at least 20 feet. The network distance of Voronoi paths are 10.52 and 9.78 mi while that of Euclidean paths are 7.36 and 7.97 mi respectively.

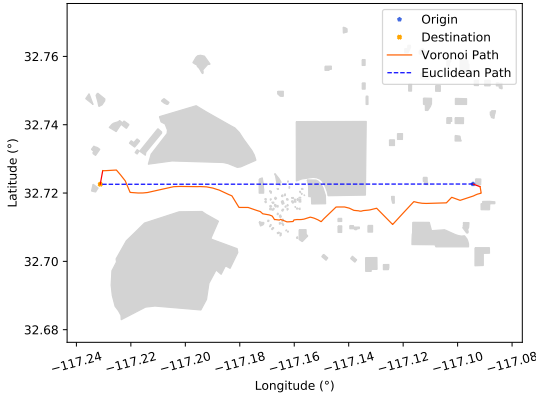
Note that Voronoi paths in this thesis are created based on just a Voronoi diagram built in the previous step. Thus, in area with sparse obstacles, the Voronoi path could not address the shortest paths between nodes because Voronoi edges are created by passing the points which bisect the straight lines between two sampling points of obstacles. This is a well known weak point of Voronoi-diagram-based methods. There are various ways to improve the Voronoi path in terms of its total length by integrating other path-planning methods. The improvement of the Voronoi paths could lead to the lower total cost of solution routes. However, it has no impact on the VRP optimization model. Thus, this thesis leaves the improvement of the Voronoi paths for future work.



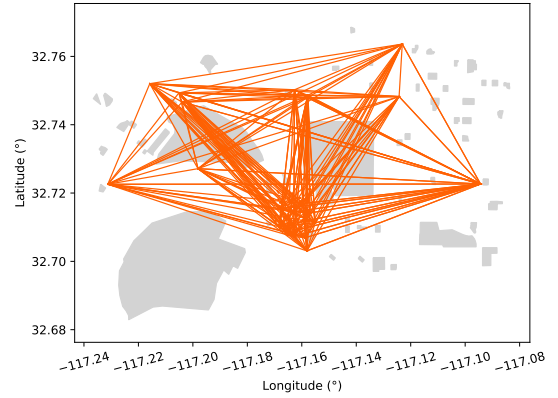
(a) Nodes of an abstract graph from a VRP



(b) Voronoi path with urban obstacles: example 1



(c) Voronoi path with urban obstacles: example 2



(d) High-level urban flight network

Figure 4.7: Voronoi paths with polygonal obstacles

After calculating all the network distances between nodes, a high-level urban flight network is created as illustrated in Figure 4.7d. Each arc has a distance calculated from a Voronoi path as its property rather than the point-to-point straight distance. A VRP uses this abstract graph as its input network topology.

This section presented a framework to create a two-layered urban flight network in Figure 4.1 with San Diego example. The following section conducts experiments to demonstrate the two-layered urban flight network created by the presented framework can address collision-free paths for a VRP optimization model.

4.3 Experiments

This section designs and conducts experiments to demonstrate the hypothesis such that

Hypothesis for RQ 3 : To address collision-free paths as an input graph of a VRP for the urban environment, a Voronoi-diagram-based method should be utilized.

To address Hypothesis for RQ 3, this section conducts two experiments which demonstrate the effects of flight networks, or input graphs, of a VRP optimization model on solution routes such that

Experiment 3 : To find which type of flight networks better represents collision-free paths in the urban environment, this experiment compares the solution routes of a point-to-point flight network with that of a two-layered urban flight network.

Experiment 4 : To address the best vehicle specification for a given scenario, this experiment conducts sensitivity analysis of maximum endurance and maximum payload capacity constraints with the integrated module with both an endurance-constrained MTVRPTW model and a two-layered urban flight network.

Specifically, Experiment 3 compares a point-to-point flight network, which is a conventional input graph of a VRP optimization model, with a two-layered urban flight network presented in this thesis. This experiment, also, conducts with an endurance-constrained MTVRPTW optimization model presented. Experiment 4 demonstrates a tendency of solution by sensitivity analysis in terms of vehicle's endurance and payload capacity to address the best options of them for given scenarios.

When addressing an optimization problem with the San Diego model, the optimization model needs three sets of inputs: depot/customer information, assumptions about sUAV operations, and vehicle performance characteristics. First, the logistics network consists of a depot and 20 delivery locations; because no company has facilities enough to cover the whole mission area, this thesis utilizes facilities of several companies in the mission

area such that the depot is a FedEx ship center, and delivery locations are the locations of FedEx, UPS, USPS, and Whole Foods in San Diego. Each delivery location has its own demand on specific time designed by time windows. The demand and time windows are stochastically created using a discrete uniform distribution which is a typical distribution in stochastic VRPs [109].

Second, this model assumes an operating altitude of 250 feet AGL, and that the time required for a sUAV to make a delivery is 10 minutes including landing and takeoff time. After a trip, each sUAV needs to swap its batteries and reload packages, which is assumed as 20-minute reloading time, including landing time. Additionally, total operating period is 8 hours—a typical daytime work shift such that the daytime operation requirement of FAR part 107 is observed.

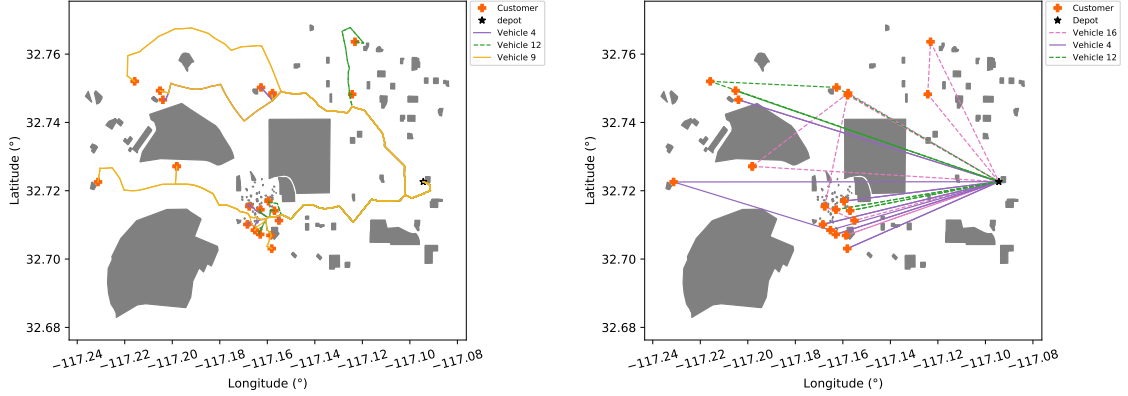
Lastly, the vehicle performance properties are operating velocity, payload capacity, and endurance which are assumed as 50 mph, which is the target speed of Amazon Prime Air [110, 111], 10 lb, and 45 minutes respectively. Under Part 107 of FARs, sUAVs are required to operate at speeds less than 100 mph, and its maximum takeoff weight should be less than 55 lb. The objective function, Eq. 3.3, has two cost coefficients: c_{fixed} and c_{ijk} . First, c_{fixed} represents the fixed cost such that

$$c_{fixed} = \text{acquisition cost of a vehicle} / \# \text{ of days in operational life}$$

Second, c_{ijk} is operating cost defined as the cost per unit distance/time when the vehicle k moves the node i to the node j in an input graph; c_{ijk} is assumed as unit cost for all simulations. These two cost terms incentivize the optimizer to select the set of routes with the minimum flight time with the minimum number of vehicles. With these conditions, all simulations are executed with Intel® Core™ i7-7700HQ processor and 32 Gb memory. Gurobi 8.0 is used as the MILP solver.

Table 4.2: Results of the effects of flight networks

Case	Flight network	Obj func.	# of sUAVs	Solution routes	CPU time (sec.)
1	Two-layered urban flight network	3857.13	3	Fig. 4.8a	2.68
2	Euclidean flight network	3772.91	3	Fig. 4.8b	2.34



(a) Solution routes with the two-layered urban flight network (b) Solution routes with the Euclidean flight network

Figure 4.8: Effects of flight networks on a solution

4.3.1 Experiment 3: Validation of a Two-Layered Urban Flight Network

To address effects of a flight network on a solution of the endurance-constrained MTRPTW model, this experiment conducts simulations of the model with two different flight networks: the two-layered urban flight network and the Euclidean flight network. The simulation results are described in Table 4.2.

Both of cases require three sUAVs, but the latter case has lower objective function value because the Euclidean distance is the shortest path between two points in two-dimensional space. However, its routes violate the restricted airspace or penetrate urban obstacles as illustrated in Fig. 4.8b. The result also reveals that the optimization model with Euclidean flight network underestimates endurance and range of sUAVs for a given scenario, and sUAVs selected by the estimated values from the model with Euclidean flight network could not operate in urban environment because of their underestimated endurance or range.

Table 4.3: Cost coefficients by different vehicle specifications

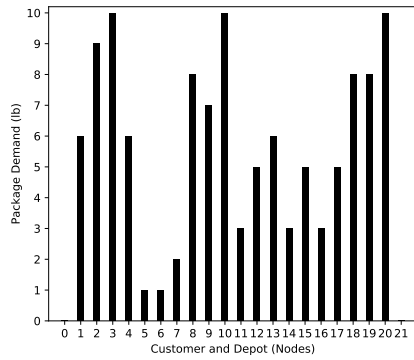
		Payload capacity (lb)		
		10	15	20
Endurance (min.)	30	1000	1200	1400
	45	1200	1400	1600
	60	1400	1600	1800

On the other hand, although the two-layered urban flight network has higher objective function, its routes are collision-free paths for the given environment which are created based on a Voronoi diagram as shown in Figure 4.8a. Therefore, this result demonstrates the fact that to address collision-free paths in an obstructed urban area, the two-layered urban flight network should be utilized with a VRP optimization model.

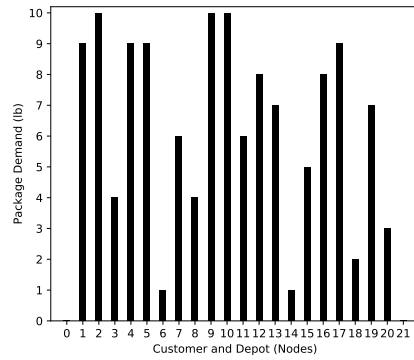
4.3.2 Experiment 4: Sensitivity Analysis

This experiment conducts sensitivity analysis of constraints of maximum endurance and maximum payload capacity of sUAVs by using the integrated module with the endurance-constrained MTRPTW optimization model and a two-layered urban flight network. This experiment runs simulations with nine combinations of endurance and payload capacity which are varied at values of 30, 45, and 60 minutes and 10, 15, and 20 lbs. To address the best combination of the endurance and payload capacity for a given scenario through sensitivity analysis, a different fixed cost coefficient, c_{fixed} , is assigned to each combination of endurance and payload as shown in Table 4.3.

Note that the value of c_{fixed} is fed by the aircraft sizing and synthesis module in the entire framework presented in this thesis. However, this chapter focuses on only the vehicle routing module. Thus, to depict the combinations of endurance and payload capacity as different sizing results, this experiment assumes that the sUAV having 30-minute endurance and 10 lb payload capacity has a c_{fixed} of 1000, and that c_{fixed} increases by 200 whenever the endurance or payload capacity is improved as shown in Table 4.3.



(a) Package demand of the first scenario



(b) Package demand of the second scenario

Figure 4.9: Two demand scenarios

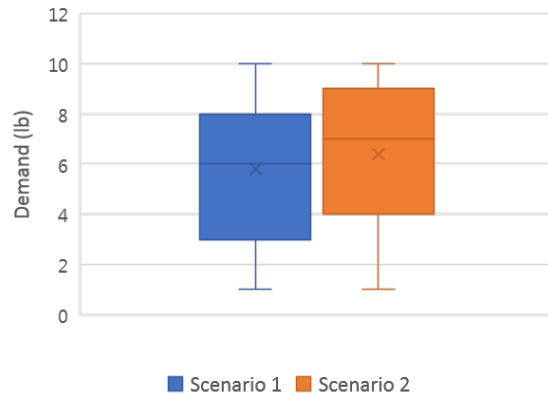


Figure 4.10: Demand distributions of two demand scenarios

In addition to vehicle performance data, the optimization model requires demand scenarios. The experiment executes simulations with two demand scenarios as shown in Figure 4.9. Each demand of nodes is created based on a normal distribution. The distribution of demands is illustrated in Figure 4.10. The distribution shows that the first demand scenario has the relatively higher possibility to carry multiple packages on a trip because its both average and median are lower than the second scenario.

Note that in this experiment, maximum CPU time for each simulation scenario is set 3600 seconds. Hence, if a model needs more than the maximum time to find a total cost optimized solution, the solver returns the best solution it finds during the limited time. Although the solution is not a total cost optimized solution, it satisfies all constraints described

Table 4.4: Results of the effects of vehicle properties and time windows with first demand set

Scenario	De- mand set	Endu- rance (min.)	Capacity (lb)	Objective function	# of sUAVs	Flight time (min.)	Opt. gap (%)	CPU time (sec.)
S13010	1	30	10	3301.13	3	301.13	0	0.90
S13015	1	30	15	3907.13	3	301.13	0	11.61
S13020	1	30	20	4501.13	3	301.13	0	10.38
S14510	1	45	10	3857.13	3	257.13	0	2.68
S14515	1	45	15	4421.00	3	221.00	0	10.57
S14520	1	45	20	5009.80	3	209.80	0.17	3600
S16010	1	60	10	4434.50	3	234.50	0	2.60
S16015	1	60	15	4981.40	3	181.40	0.45	3600
S16020	1	60	20	3767.82	2	167.82	0.97	3600
S23010	2	30	10	4301.13	4	301.13	0	0.99
S23015	2	30	15	5101.13	4	301.13	0	1.11
S23020	2	30	20	5901.13	4	301.13	0	5.27
S24510	2	45	10	5061.77	4	261.77	0	4.27
S24515	2	45	15	5842.08	4	242.08	0	24.31
S24520	2	45	20	5002.20	3	202.20	0	37.58
S26010	2	60	10	5841.14	4	241.14	0	1.17
S26015	2	60	15	6623.94	4	223.94	0	6.40
S26020	2	60	20	3795.92	2	195.92	0	19.70

in Section 3. When the returned solution is not an optimum, the solver returns a non-zero optimality gap² which is defined by

$$\text{Optimality gap} = \frac{|\text{bound value} - \text{objective function value}|}{|\text{objective function value}|} \quad (4.1)$$

With these assumptions, sensitivity analysis of the integrated module is executed. To address different combinations of vehicles properties and demand sets, 18 simulation scenarios are generated. Each simulation scenario is named by concatenating demand set,

²<http://www.gurobi.com/documentation/8.1/refman.pdf> (accessed 22 January 2019)

endurance, and payload capacity. For instance, S13010 is conducted with first demand set and vehicles that have 30-minute endurance and 10 lb payload capacity.

The results are summarized in Table 4.4. Note that flight time shows the total time of all sUAVs to calculate operating cost. First of all, by comparing objective function values of scenarios with a same demand set, the most cost-efficient vehicle for the demand set can be selected. Hence, in the first demand scenario, vehicles with 30-minute endurance and 10 lb payload capacity are the best option, whereas in the second scenario, vehicles with 60-minute endurance and 20 lb payload capacity are the most preferred choice. This is the fundamental result of sensitivity analysis in this experiment.

Second, to address an effect of endurance, three scenarios such as S13010, S14510, and S16010 or S23015, S24515, and S26015, which have the same both demand set and payload capacity condition, need to be compared together. For instance, when comparing S23015, S24515, and S26015, flight time decreases as endurance increases. This result is expected because as endurance increases, the probability that a sUAV can visit next customer without returning to a depot increases. This effect can also be observed in other five cases. In addition, this tendency also is observed in Experiment 2 in the previous chapter.

Third, the effect of payload capacity is shown by analyzing three scenarios such as S14510, S14515, and S14520 or S26010, S26015, and S26020, which have the same both demand set and endurance condition. Note that when a vehicle has 30-minute endurance, even if payload capacity increases, flight time is still the same. This implies that it is physically impossible to obtain a better solution by visiting more delivery locations on a route due to endurance constraints. When comparing S14510, S14515, and S14520, flight time decreases as payload capacity increases. This is expected because an opportunity that a sUAV can visit more delivery locations on one route grows. Three other cases about analyzing payload capacity effect have a similar result.

Note that endurance and payload capacity have an effect on a route, not a journey.

Longer endurance and larger payload capacity create a delivery plan with fewer sUAVs. However, this tendency occurs due to a decrease of the number of routes, not a increase of the reuse of sUAVs. Mathematically, the reusability design variables, σ_{ij} , have no direct relation with endurance or payload capacity (See Equation (3.10)-(3.13)).

These results demonstrate that the endurance-constrained MTRPTW model works well with the two-layered urban flight network of San Diego for the given scenarios; moreover these results show how trade studies can be conducted to determine vehicle parameters by utilizing delivery scenarios. Note that the two-layered urban flight network is created by the framework presented in this chapter. Because the framework deals with each step as a block box function, other urban models can be created by feeding point cloud data into the first step of the framework. Additionally, if the depot/customer information of the urban models are available, the endurance-constrained MTRPTW model also can be utilized with the two-layered urban flight networks for other urban models.

4.4 Conclusions

This chapter initiated from the research question such that

Research Question 3 (RQ 3) : What type of an input graph of a VRP model best represents collision-free paths in the urban environment for sUAS-based delivery systems?

To address this research question, a literature review of path-planning methods was conducted. There were two main observations from the review: the first observation was that road map methods can efficiently represent the urban environment to address collision-free paths for an input graph of a VRP optimization model. The second observation was that the Voronoi diagram, which is a road map method, can be efficiently created for a large-scale obstructed area. These observations inspired the hypothesis such that

Hypothesis for RQ 3 : To address collision-free paths as an input graph of a VRP for the urban environment, a Voronoi-diagram-based method should be utilized.

To prove Hypothesis for RQ 3, a framework to create a two-layered urban flight network based on a Voronoi diagram was presented. Then, experiments were conducted. Experiment 3 demonstrated that to address collision-free paths in an obstructed urban area, the two-layered urban flight network created by the framework should be utilized with a VRP optimization model. Furthermore, Experiment 4 which is sensitivity analysis also confirmed that the endurance-constrained MTVRPTW model works well with the two-layered urban flight network for the given scenarios. These results lead to a conclusion that Hypothesis for RQ 3 is substantiated.

CHAPTER 5

IMPLEMENTATION OF A SIZING AND SYNTHESIS PROCESS FOR SMALL FIXED-WING VTOL UAVS

To address an aircraft design process, this chapter presents a notional sizing and synthesis method for sVTOL UAVs. This thesis previously assumed that tiltrotors which uses 8 rotors for vertical flight and 4 rotors for forward flight are used for a given delivery mission. Thus, the configuration of the tiltrotor is used to implement a sizing and synthesis process to address the research question such that

Research Question 1 (RQ 1) : How can a set of main components of small fixed-wing VTOL UAVs as a result of a sizing and synthesis process be obtained?

Note that the intent of this chapter is to implement a notional sizing and synthesis method for sVTOL UAVs of an early design step rather than a preliminary design tool providing detailed UAV design parameters with aircraft configurations. The main purpose of the sizing and synthesis method presented in this chapter is to be used as a simple black box for a UAV design module in the framework for concurrent UAV design and routing for urban delivery systems in Figure 1.21. If more detailed sizing results of small fixed-wing VTOL UAVs are needed, other sizing and synthesis processes having the same interface can be used for the framework for concurrent UAV design and routing.

Section 5.1 develops a hypothesis for Research Question 1. Section 5.2 presents a component-based notional sizing and synthesis process for sVTOL UAVs. Then, Section 5.3 executes experiments to substantiate the hypothesis for Research Question 1.

5.1 Development of Hypothesis

The traditional sizing and synthesis method mentioned in Chapter 1 focuses on small fixed-wing aircraft. However, to address a sizing and synthesis method for sVTOL UAVs, vertical flight capacity should be considered in the method. In order to design a small fixed-wing VTOL UAV, Kamal and Ramirez-Serrano [112] proposed a design methodology using mathematical formulations to find an optimal design point in a design chart of constraint analysis. Then, they conducted a preliminary design for a box wing tiltrotor UAV, but their process did not address available real components to obtain the designed vehicle. Tyan et al.[113] presented a sizing and synthesis method for a small fixed-wing VTOL UAV consisting three steps: initial sizing base on mathematical formulations, selecting actual components, and resizing to update aircraft design parameters based on selected components. However, they did not specifically mentioned how the components are finally selected.

There are other sizing method focusing on designing the propulsion system of small UAVs because the propulsion system of small UAVs accounts for up to about 60 % of their weight [114, 28]. To address the propulsion system sizing problems, parametric design methods can be utilized, which use models regressed by actual market component data to describe the relation between performance factors and weight for each part [114, 115, 116]. Gur and Rosen [114] presented a parametric design method including electric motor and battery models while Ampatis and Papadopoulos [115] and Bershadsky et al.[116] proposed a parametric method with electric motor, electronic speed controller, battery and propeller models. However, these parametric design methods work with only electrical rotors or multirotors, and they do not address specific components as a sizing result.

Additionally, to address battery's electrical properties, Stepaniak et al.[117] presented a mathematical model for an electric system of a sUAV. They addressed battery's internal resistance occurring the loss dissipated for high-power applications. Then, Ampatis and Papadopoulos [115] extended the battery model by adding battery configuration such as

the number of cells/packs.

From this literature review, there are two observations: the first observation is that for small fixed-wing VTOL UAVs, non-parametric approaches [112, 113] should be utilized because the parametric approaches [114, 115, 116] works with only single/multi-rotor aircraft. The second observation is that a sizing and synthesis process for a small VTOL UAV should address its propulsion system specifically. These observations inspire the hypothesis such that

Hypothesis for RQ 1 : To obtain a set of components of a propulsion system for small fixed-wing VTOL UAVs, a component-based sizing and synthesis process can be utilized.

Both non-parametric methods [112, 113] focus on designing a small fixed-wing VTOL UAV system with a specific configuration of the UAV. However, to address selecting components, a sizing and synthesis module implemented by this thesis are based on the sizing and synthesis method presented by Tyan et al. [113]. Additionally, to address detailed battery configurations such as the number of cells/packs, the battery model of [115] is integrated. The following section presents the implemented sizing and synthesis module.

5.2 Aircraft Sizing and Synthesis Process for a VTOL UAV

The implemented sizing and synthesis process for a small fixed-wing sVTOL UAV deals with a notional design concept without a detailed vehicle configuration. However, to address the propulsion system design, a notional configuration of a tilting-rotor VTOL which uses eight motors for vertical flight and four motors for forward flight is used to make the process. The sizing and synthesis process includes both constraint analysis and weight analysis. The overall process is shown in Figure 5.1. First, an optimal design point is selected by constraint analysis considering the fixed-wing aircraft performance such as cruise, climb, and stall. Next, hovering capacity and forward flight capacity are addressed

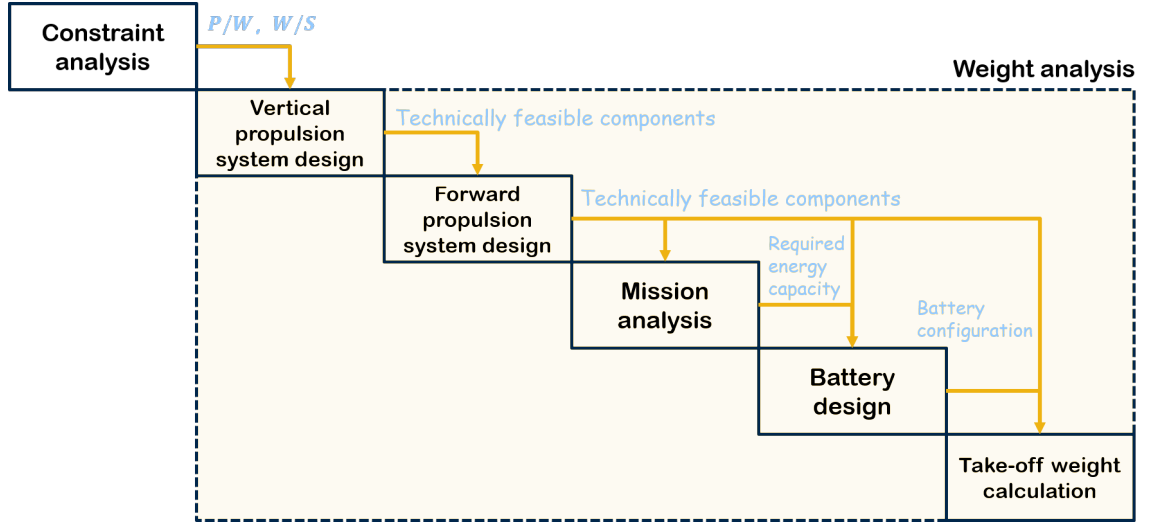


Figure 5.1: Process of the developed fixed-wing VTOL UAV sizing module

to find feasible electrical motors and propellers. Note that in the implemented sizing and synthesis process, after finding technically feasible components for hovering capacity, the process finds technically feasible components for forward flight capacity. This sequence is computationally efficient because the process to find technically feasible components for forward flight capacity requires more computation power. Then, the required energy capacity is calculated by mission analysis. Based on the energy capacity, battery's configuration is determined. Lastly, take-off weight is calculated.

5.2.1 Mission Profile

A mission profile describes what the goal of the aircraft is and how the vehicle operates. A mission profile consists of sequential mission segments to complete an objective [24]. Mission segments, or legs, describe a specific part of the mission such as warm-up and takeoff, climb, cruise, loiter, descent, and landing. MIL-STD 3013 [118] provides typical mission profiles for a variety of types of missions with descriptions of mission segments. Each mission segment includes specific conditions in terms of speed, altitude, etc. The mission profiles should include all important characteristics or conditions of the mission because constraints analysis and weight analysis are conducted based on each mission segment.

5.2.2 Constraint Analysis

Constraint analysis is utilized to find feasible design space in terms of performance constraints. In order to describe each constraint, Mattingly [20] presented the master equation describing flight performance constraints as both thrust/power loading and wing loading. In this section, a simplified variant of the master equation presented by Gudmundsson [119] is used for constraint analysis.

To address constraint analysis for propeller-driven aircraft, each flight performance constraint should be described by power loading and wing loading. Power loading can be derived from thrust loading such as

$$\left(\frac{P}{W}\right) = \frac{(T/W)V}{\eta_p} \quad (5.1)$$

where (P/W) is power loading, (T/W) is thrust loading, η_p is the propeller efficiency, and V is the airspeed. The power of electric motors is independent from altitude, thus normalizing power to sea level is not required [113].

In order to deal with flight performance of fixed-wing mode, three types of performance constraints, cruise, climb, and stall, are considered. First, cruise constraint can be written by

$$\left(\frac{T}{W}\right)_{cruise} = \frac{q \cdot C_{D0}}{(W/S)} + \frac{k \cdot (W/S)}{q} \quad (5.2)$$

where q is the dynamic pressure, C_{D0} is the drag coefficient at zero lift, and k is the aerodynamic coefficient. Second, climb performance constraint can be described by

$$\left(\frac{T}{W}\right)_{climb} = \frac{V_v}{V_{roc}} + \frac{q \cdot C_{D0}}{(W/S)} + \frac{k \cdot (W/S)}{q} \quad (5.3)$$

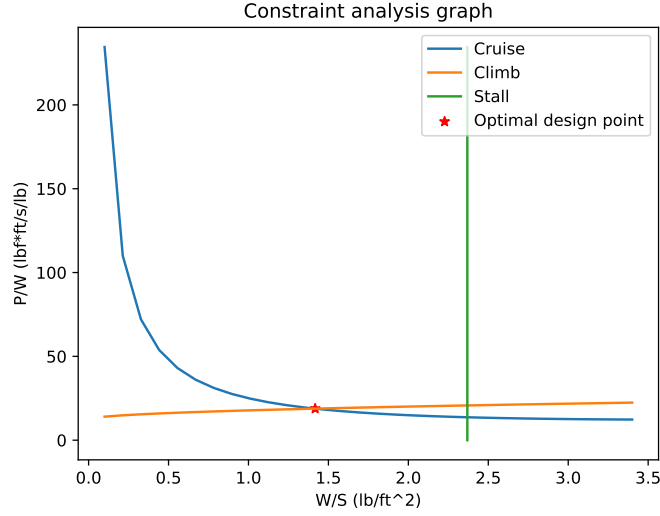


Figure 5.2: Constraint analysis graph consisting of cruise, climb, and stall

where V_v is the vertical speed, and V_{roc} is the best rate-of-climb airspeed defined by

$$v_{roc} = \sqrt{\frac{2}{\rho} \left(\frac{W}{S} \right)} \sqrt{\frac{k}{3C_{D0}}} \quad (5.4)$$

Last, stall constraint can be written by

$$\left(\frac{W}{S} \right)_{stall} = q_{stall} \cdot C_{L,max} \quad (5.5)$$

where q_{stall} is the dynamic pressure at the aircraft stall speed, and $C_{L,max}$ is the maximum lift coefficient. Although VTOL aircraft reaches the stall speed, it does not stall. However, it is a substantial design speed for the transition phase [113]. An example of constraint analysis graph is shown in Figure 5.2. The optimal design point can be determined as a point requiring minimum power loading in feasible design space satisfying all constraints.

5.2.3 Weight Analysis

The implemented weight analysis consists of five steps as previously illustrated in Figure 5.1: vertical propulsion system design, forward propulsion system design, mission analysis,

battery design, and take-off weight calculation. The following subsections describe each step.

Step 1: Vertical Propulsion System

The developed weight analysis process is a components-driven method, which finds feasible pairs of components for given constraints and then evaluates the convergence of the aircraft takeoff weight. The first step is to find motor and propeller pair satisfying hovering capacity. Based on the momentum theory, the power required to hover is defined by [120]

$$P_h = T \sqrt{\frac{T}{2\rho A}} = \frac{T^{3/2}}{\sqrt{2\rho A}} = \frac{W^{3/2}}{\sqrt{2\rho A}} \quad (5.6)$$

where T is the thrust required to hover, W is the aircraft weight, ρ is the air density, and A is the disk area. Furthermore, for multicopters consisting of n rotors, the ideal power required to hover for each rotor can be defined by

$$P_{h,(1/n)} = \frac{(W/n)^{3/2}}{\sqrt{2\rho(A/n)}} \quad (5.7)$$

For an actual multirotor design, thrust-to-weight ratios of $1.5 \sim 2.0$ is used [121]. Manufacturers of motors for sUAVs provide test results of thrust/power by throttle settings. Thus, with desired thrust-to-weight ratios, proper motors can be selected by using the results. Furthermore, the Figure of Merit (FM) is commonly used to estimate practical power, which is defined by [120]

$$FM = \frac{P_{ideal}}{P_{actual}} < 1 \quad (5.8)$$

Using these values, available vertical propulsion components are can be selected from the component database made up of data sheets for motor-propeller performance provided by manufacturers.

Step 2: Forward Propulsion System

To address forward flight performance, the required thrust needs to be calculated, which is defined by the drag of a given flight condition such that [122]

$$T_R = D = \frac{1}{2}\rho V^2 SC_{D_0} + \frac{2kS}{\rho V^2} \left(\frac{W}{S} \right)^2 \quad (5.9)$$

Then, the required power can be written by [122]

$$P_R = T_R V \quad (5.10)$$

In order to evaluate the feasibility of propellers for the forward flight, thrust at target velocity should be obtainable by the propulsion system. Thrust of a propeller relies on propeller's revolutions per minute (rpm) and velocity. Also, the propulsion system should support power transmitted by a shaft, P_{shaft} , at the conditions. In order to obtain these values, experiments such as a wind tunnel test [123] or a Computational Fluid Dynamics (CFD) analysis [124] can be utilized. Various propeller performance test results are obtainable in [125] and [126]. This thesis uses the test results to create the component database to determine actual propulsion system parts such as propellers, electric motors, and batteries.

After calculating the required thrust/power, each propeller in the component database is evaluated to determine whether or not it can produce the thrust/power at cruise speed of 50 mph previously assumed in Section 4.3. Figure 5.3 shows an example to find available thrust of a propeller by using regression models for each propeller performance in terms of thrust, velocity, and rpm. In this plot, feasible rpm values of the propeller can be identified by the required thrust and the target velocity. For instance, if the required thrust is 10 lbf, then the propeller requires at least 10000 rpm to get the thrust at speed of 50 mph.

P_{shaft} , also, can be determined by regression models of Figure 5.4 in the same manner. For example, if a given rpm is 10000 rpm, P_{shaft} is about $2 \text{ lbf} * \text{ft/s}$ at speed of 50

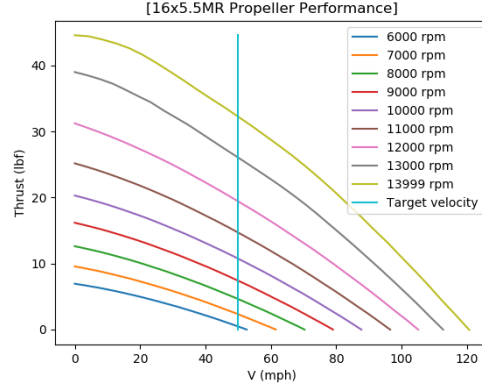


Figure 5.3: Regression models of propeller performance: thrust, velocity, and rpm

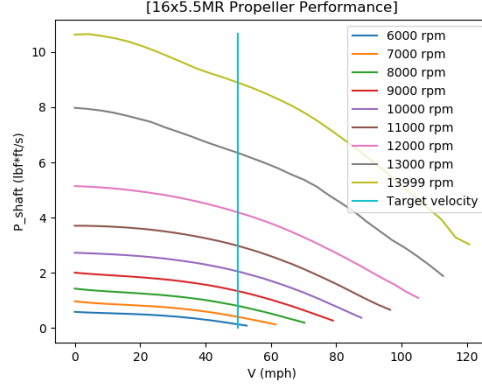


Figure 5.4: Regression models of propeller performance: P_{shaft} , velocity, and rpm

mph. The selected P_{shaft} value should be supported by electric motors of the propulsion system. In this step, all the available propellers are selected because the performance could be changed by the combination of a motor and a propeller.

Manufacturers of propulsion system parts for sUAVs may provide the performance data. If it is available, the input electric power transferred into a motor, P_{in} , can be directly considered instead of P_{shaft} . P_{in} is the electric power which should be supported by electric batteries. For sUAVs, using BrushLess DC (BLDC) motors, which remove the needs for brushes used by older DC motors, is preferred because of its converting efficiency from electrical energy to mechanical energy [116]. BLDC motors are controlled by Electronic Speed Controller (ESC). The ESC receives throttle setting values through Pulse Width Modulation (PWM) signals, and then it controls the rpm of motors by the percentage duty

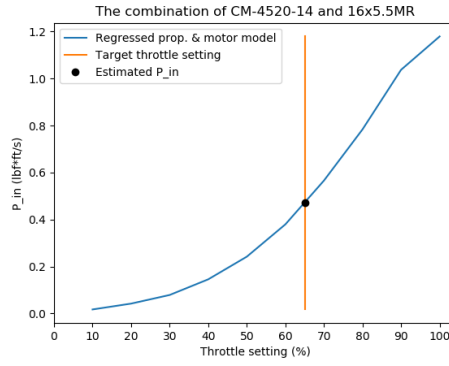


Figure 5.5: Regression models of motor and propeller performance: throttle setting and P_{in}

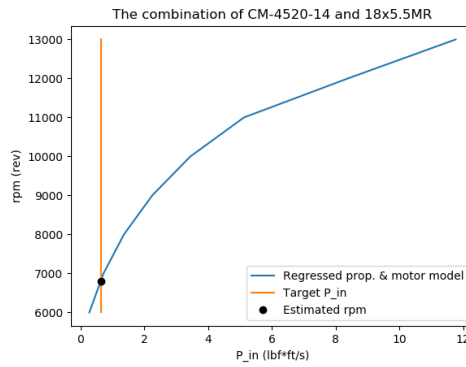


Figure 5.6: Regression models of motor and propeller performance: P_{in} and rpm

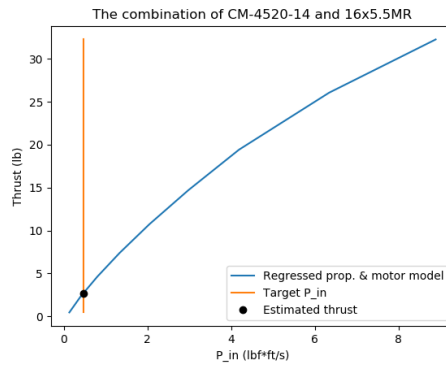


Figure 5.7: Regression models of motor and propeller performance: P_{in} and thrust

cycle [127].

In the motor and propeller performance data provided by manufacturers, available thrust/power is measured for various throttle values. Based on a target throttle value, 65 % is selected from [128, 129], P_{in} can be determined as illustrated in Figure 5.5. The blue

curve in Figure 5.5 is regression model based on motor-propeller performance data. Then, the rpm of motor is estimated by P_{in} as seen in Figure 5.6 in the same manner. Lastly, the available thrust also can be calculated by P_{in} using the regression model as illustrated in Figure 5.7. Thus, feasible combinations of a motor and a propeller which produce more thrust than required thrust can be selected.

Step 3: Mission Analysis

The purpose of mission analysis in weight analysis in this thesis is to calculate the required battery capacity. For each mission segment, the required P_{in} should be calculated because it is changed by the flight state. Package delivery missions can be simply modeled by combinations of take-off, cruise, and landing. Thus, this thesis uses the hovering power for take-off and landing mission segments and the cruise power for cruise mission segments. Then, the required battery capacity for each mission segment is determined by multiply the required P_{in} by time for a mission segment. Lastly, the required battery capacity for a given mission is calculated by summing all the required battery capacity for each mission segment.

Step 4: Battery Design

The designed electrical battery should have more capacity than the required battery capacity for a given mission. An electrical battery consists of multiple cells. To address the configurations of cells, the battery model of Ampatis and Papadopoulos [115] is used. The configurations of cells rely on the design of UAV's power system; the input voltage of a motor can be supported by the batteries, or a DC-DC converter can be used to support the high input voltage of a motor. This thesis assumes that the motor input power is directly supported by the batteries. Then, based on the motor input voltage, the battery pack size is determined, which consists of cells grouped in a serial arrangement. By stacking the battery packs, the battery capacity is adjusted.

Step 5: Take-off Weight Calculation

After mission analysis, the weight of electric motors, propellers, and battery's configuration are determined for each feasible combinations of a motor and a propeller. If there exist multiple feasible options, the option having the minimum weight is chosen. In order to address the convergence condition of sizing iterations, a sizing equation is utilized, which can be written by

$$W_{to} = \frac{W_{propulsion} + W_{battery} + W_{payload}}{(1 - WF_{structure} - WF_{avionics} - WF_{subsystem})} \quad (5.11)$$

where W is the actual weight and WF is the weight fraction. Since in previous steps, the components of motors, propellers, and batteries are selected, the actual weight of them can be addressed in the sizing equation. For the weight fractions, $WF_{structure} = 0.33$, $WF_{avionics} = 0.05$, and $WF_{subsystem} = 0.1$ are used [121].

This section presented how a component-based sizing and synthesis process of this thesis is implemented to obtain a set of components of a propulsion system for small fixed-wing VTOL UAVs. The following section executes experiments to validate the implemented sizing and synthesis process.

5.3 Experiments

This section designs and conducts experiments to demonstrate the hypothesis such that

Hypothesis for RQ 1 : To obtain a set of components of a propulsion system for small fixed-wing VTOL UAVs, a component-based sizing and synthesis process can be utilized.

To address the hypothesis, a component-based sizing and synthesis process is implemented, which can provides a set of components based on component database. To validate the implemented process, two experiments are executed such that

Table 5.1: Assumptions for sizing and synthesis process

Group	Parameter	Value	Reference
Aerodynamics	T/W for vertical flight	2.0	[121]
	L/D	5.6	[121]
	e	0.65	[24]
	AR	4.406 ¹	
	C_{D_0}	0.04	[113]
Propulsion	Battery-specific energy (Wh/kg)	253	[121]
	Battery efficiency (%)	75	[121]
	# of hover motors	8	
	# of cruise motors	4	
	T/W hover	2	[121]
	Throttle value for cruise (%)	65	[128, 129]
Mass	$WF_{structure}$	0.33	[121]
	$WF_{avionics}$	0.05	[121]
	$WF_{subsystem}$	0.1	[121]

Experiment 5 : This experiment conducts a sizing and synthesis process for a single-stop delivery mission to validate the implemented sizing and synthesis process.

Experiment 6 : This experiment executes a sizing and synthesis process for a two-stop delivery mission to validate the implemented sizing and synthesis process.

The main purpose of the two experiments is to confirm whether the implemented sizing and synthesis process provides a reasonable and feasible design values for the given missions. Specifically, Experiment 5 conducts a sizing task to find UAVs design parameters such as P/W , W/S , and take-off weight for a single-stop delivery mission which describes that a small fixed-wing VTOL UAV delivers a package to a customer. Experiment 6 executes another sizing task to find design parameters for a two-stop delivery mission which a small fixed-wing VTOL UAV delivers two packages to two different locations on a trip.

To address a sizing and synthesis process, assumptions in terms of aerodynamics, propulsion and mass are required. Table 5.1 shows common assumptions for two experi-

¹This value is calculated by an image of the Wingcopter <https://wingcopter.com/> (accessed 18 February 2019)

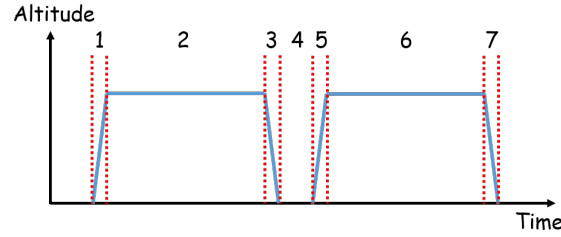


Figure 5.8: Mission profile of a single-stop delivery

Table 5.2: Mission segments of a single-stop delivery

Segment No.	Mode	Description
1	Take-off	Take off and climb up to 250 ft during 60 sec
2	Cruise	Cruise 10 mi at speed of 50 mph
3	Landing	Land on the ground during 60 sec
4	Delivery	Add 60-sec hovering flight as a margin
5	Take-off	Take off and climb up to 250 ft during 60 sec
6	Cruise	Cruise 10 mi at speed of 50 mph
7	Landing	Land on the ground during 60 sec

ments. Because the implemented process uses specification and performance data of each component of propulsion systems from component database to select final components, the factors such as propulsion efficiency depending on the selected component are determined during the process of sizing and synthesis.

5.3.1 Experiment 5: Single-Stop Delivery Mission

In a single-stop delivery mission, a 5-lb package is delivered to a location which is ten miles away from a depot. This is a typical urban delivery mission of sUAVs; in most U.S. urban areas, 10- to 15-mile delivery radius is enough to cover the areas [121]. The mission profile of this delivery mission is illustrated in Figure 5.8, and each mission segment is described in Table 5.2.

In the initial constraint analysis, power loading P/W , and wing loading, W/S , of the design point are $18.82 \text{ lbf} \cdot \text{ft/s/lb}$ and 1.42 lb/ft^2 as shown in Figure 5.9. During the sizing process, take-off weight is updated until it is converged. The iteration result is de-

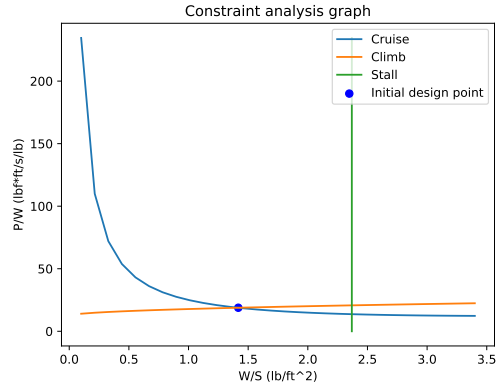


Figure 5.9: Initial design point of a single-stop mission

Table 5.3: Mission profile of a single-stop delivery

Iter.	$W_{to,in}$ (lb)	$W_{propulsion}$ (lb)	$W_{battery}$ (lb)	$W_{to,calculated}$ (lb)	ΔW_{to} (%)
1	40	5.59	5.16	30.29	24.29
2	30.29	5.89	3.97	28.57	5.65
3	28.57	5.59	3.97	28.00	2.02
4	28.57	5.59	3.97	28.00	0

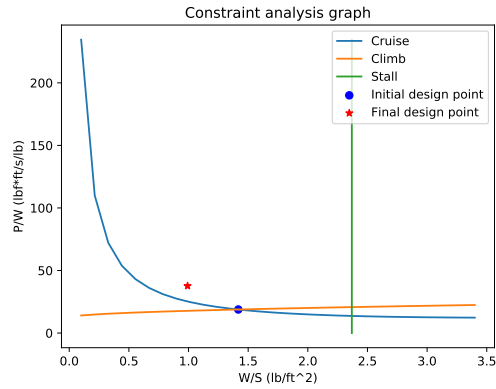


Figure 5.10: Final design point of a single-stop mission

scribed in Table 5.3. This result is in between the two design results of the baseline aircraft and the advanced aircraft for a single-package delivery mission presented in [121]. After iterations, the updated value of power loading and wing loading is $37.69 \text{ lb} \cdot \text{ft}/\text{s}/\text{lb}$ and $0.99 \text{ lb}/\text{ft}^2$, which is in feasible design space as illustrated in Figure 5.10. Accord-

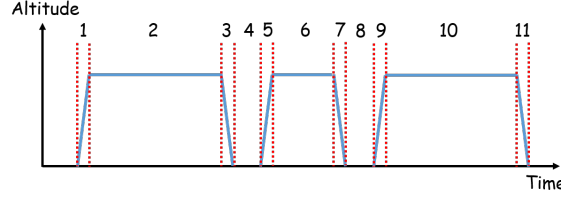


Figure 5.11: Mission profile of a two-stop delivery

Table 5.4: Mission segments of a two-stop delivery

Segment No.	Mode	Description
1	Take-off	Take off and climb up to 250 ft during 60 sec
2	Cruise	Cruise 10 mi at speed of 50 mph
3	Landing	Land on the ground during 60 sec
4	Delivery	Add 60-sec hovering flight as a margin
5	Take-off	Take off and climb up to 250 ft during 60 sec
6	Cruise	Cruise 5 mi at speed of 50 mph
7	Landing	Land on the ground during 60 sec
8	Delivery	Add 60-sec hovering flight as a margin
9	Take-off	Take off and climb up to 250 ft during 60 sec
10	Cruise	Cruise 10 mi at speed of 50 mph
11	Landing	Land on the ground during 60 sec

ing to [113], appropriate values of P/W for propeller-driven aircraft exists within 2-30 W/N . Since $37.69 \text{ lbf} \cdot \text{ft/s/lb} = 11.49 \text{ } W/N$, the result is within the acceptable range. Thus, these results demonstrate the fact that the implemented sizing and synthesis process for small fixed-wing VTOL UAVs provides feasible and reasonable design parameters of P/W , W/S , and take-off weight.

5.3.2 Experiment 6: Two-Stop Delivery Mission

The second scenario is designed to address a multi-package delivery by a sUAV. The scenario is to deliver two packages to different places. The first delivery location is 10 miles away from the depot, and the second location is 5 miles away from the first location, which is also 10 miles away from the depot. At each location, a 5-lb package is delivered by a sUAV. During each service time, 60-second hovering flight is added as a margin. For a

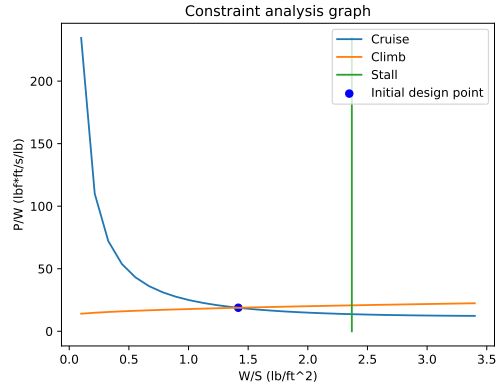


Figure 5.12: Initial design point of a two-stop mission

Table 5.5: Mission profile of a two-stop delivery

Iter.	$W_{to,in}$ (lb)	$W_{propulsion}$ (lb)	$W_{battery}$ (lb)	$W_{to,calculated}$ (lb)	ΔW_{to} (%)
1	40	6.81	6.75	42.95	7.38
2	42.95	7.11	6.74	45.87	6.79
3	45.87	7.11	7.14	46.63	1.66
4	45.87	7.11	7.14	46.63	0

sizing process, a mission profile in Figure 5.11 is created by the scenario, and each mission segment is specified in Table 5.4.

In the initial constraint analysis, a design point with power loading of $18.82 \text{ lbf} \cdot \text{ft/s/lb}$ and wing loading 1.42 lb/ft^2 is obtained, which is in feasible design space as illustrated in Figure 5.12. During the sizing iterations, the take-off weight of the sUAV is converged to 48.16 lb which is less than the maximum take-off weight, 55 lb, under Part 107 as shown in Table 5.5. The final design point has power loading of $30.09 \text{ lbf} \cdot \text{ft/s/lb}$ (9.18 W/N) which is in acceptable range of [113] and wing loading of 1.65 lb/ft^2 . The design point also is in feasible space as illustrated in Figure 5.13. These results also substantiate the fact stated in previous experiment that the implemented sizing and synthesis process for small fixed-wing VTOL UAVs provides feasible and reasonable design parameters of P/W , W/S , and take-off weight.

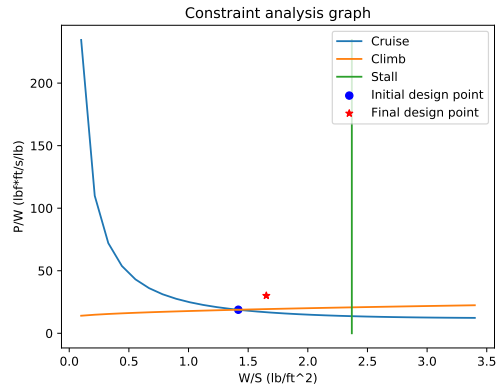


Figure 5.13: Final design point of a two-stop mission

5.4 Conclusions

This chapter started from the research question such that

Research Question 1 (RQ 1) : How can a set of main components of small fixed-wing VTOL UAVs as a result of a sizing and synthesis process be obtained?

To address this research question, a literature review of sizing and synthesis methods for sV-TOL UAVs was executed. From the literature review, there were two observations: the first observation was that non-parametric approaches should be used to address small fixed-wing VTOL UAVs. The second observation was that the propulsion systems should be mainly addressed in the sizing and synthesis process for a small VTOL UAV. These observations inspired the hypothesis such that

Hypothesis for RQ 1 : To obtain a set of components of a propulsion system for small fixed-wing VTOL UAVs, a component-based sizing and synthesis process can be utilized.

To deal with Hypothesis for RQ 1, a component-based sizing and synthesis process for small fixed-wing VTOL UAVs was implemented. The sizing and synthesis process has five steps: first, feasible design space is bounded by constraint analysis, and the initial design point is selected from the feasible design space. Second, in weight analysis, the

required power for vertical/forward flight is calculated, and the candidates of the vertical propulsion system are selected. Then, using regression models created by the component database, the candidates of the forward propulsion system are chosen. Third, mission analysis is conducted to calculate the required battery capacity for a given mission. Fourth, the configuration of batteries is determined by the required battery capacity. Lastly, the convergence of iterations is checked by the sizing equation to find take-off weight.

To prove Hypothesis for RQ 1, Experiment 5 and 6 were executed with two delivery scenarios: a single-/two-stop delivery mission respectively. The former scenario represents a single-package delivery UAV while the latter scenario describes a multi-package delivery UAV. To address the validation of the implemented sizing and synthesis process, the result was compared with that of the existing researches [113, 121]. The results demonstrated the fact that the implemented sizing and synthesis process for small fixed-wing VTOL UAVs provides feasible and reasonable design parameters of P/W , W/S , and take-off weight. Therefore, the results show that Hypothesis for RQ 1 is substantiated.

CHAPTER 6

A FRAMEWORK FOR CONCURRENT UAV DESIGN AND VEHICLE ROUTING PROBLEMS FOR URBAN DELIVERY

To address a framework for concurrent UAV design and routing for urban delivery systems, Chapter 3 presented an endurance-constrained MTRPTW optimization model which represents operations of sUAS-based delivery systems. Chapter 4 developed a framework to create a two-layered urban flight network which is an input graph of the optimization model presented in Chapter 3 to address collision-free paths in the urban environment. Chapter 4 implemented a component-based sizing and synthesis process for small fixed-wing VTOL UAVs.

To deal with a converged solution of UAV design module and vehicle routing module, this thesis previously determined to utilize a FPI method. The coupled factors in a FPI were vehicle specification and mission profiles. Based on the observation, this chapter addresses the research question such that

Research Question 4 : How can the small VTOL UAV design module and the VRP module be integrated with a FPI method to obtain a converged solution?

Using a FPI method to solve an MDO problem is a common approach. Thus, this chapter does not address a novel approach based on the FPI. Note that the intent of this chapter is to address the extension of a sizing and synthesis process by adding a mission-planning notion for sUAS-based urban delivery missions based on a FPI method. This chapter focuses on how to integrate both the UAV design module and the VRP module with a FPI method presented in Section 2.1.

Section 6.1 develops a hypothesis for Research Question 4. Section 6.2 presents a framework for concurrent UAV design and routing for urban delivery. Then, Section 6.3

conducts experiments to substantiate the hypothesis with the presented framework for concurrent UAV design and routing for urban delivery.

6.1 Development of Hypothesis

This thesis uses a FPI method described in Section 2.1 to find a converged solution of an MDO problem for a concurrent UAV design and vehicle routing problem. To address a FPI method, three parts should be specified such that

- What are subsystems which should be addressed for a converged solution?
- What are coupled variables between subsystems?
- How can convergence of a solution be evaluated?

For the first part, the sizing and synthesis process presented in Chapter 5 and the VRP optimization formulations presented in Chapter 3 can be modeled as subsystems of a FPI as stated previously.

For the second part, the vehicle specification and mission profiles can be designed as coupled parameters as shown in Section 2.1. The vehicle specification is determined by the UAV design module as numerical values in each iteration. Thus, the vehicle design parameters can be used directly as coupled variables. In this thesis, a main issue of a designed vehicle is its flight capacity. There are three types of constraints to describe vehicle's flight capacity: maximum range [130, 131], maximum endurance [82], and maximum energy capacity [11, 132]. Because endurance/range of a sUAV is changed by flight conditions, tracking energy of electric batteries can provide more precise information for creating solution routes. To utilize maximum energy capacity constraints to describe flight capacity, maximum battery capacity, the required power for forward flight, and the required power for vertical flight are required [11, 132]. Note that to address the vehicle design parameters for the vehicle routing module, the optimization model for mission planning should include energy concepts.

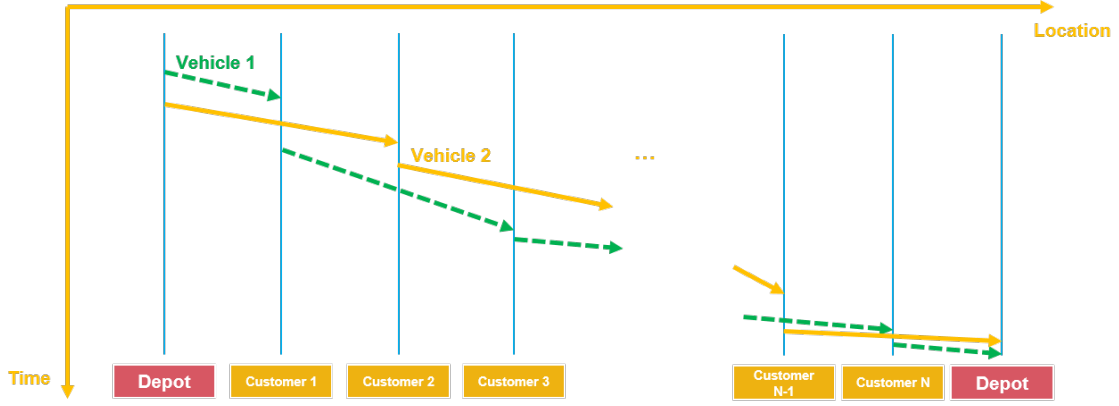


Figure 6.1: A notional example of a solution of the vehicle routing module

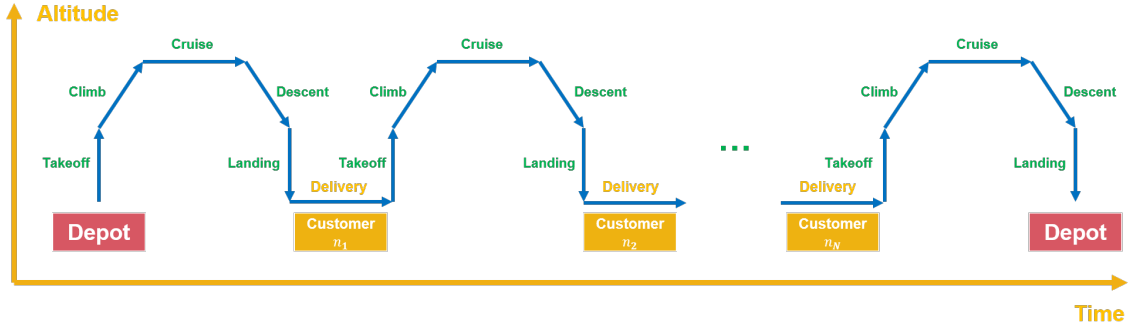


Figure 6.2: A dynamic mission profile

On the other hand, a solution from the vehicle routing module is a set of routes that are described in a time-space network. Figure 6.1 shows a notional time-space network; x-axis shows locations of nodes such as depots and customers while y-axis presents time. Each trip or journey of vehicles can be displayed in the time-space plot as shown in Chapter 3.

To address a mission profile as a coupled variable in the FPI, this thesis uses the worst trip in terms of required flight time because if a vehicle can move along the worst route for a delivery mission, the vehicle also can move along any route in the solution. During the FPI process, the worst route may be changed in each iteration. Hence, this thesis calls it a dynamic mission profile.

The dynamic mission profile for the package delivery can consist of takeoff, climb, cruise, descent and landing segments as illustrated in Figure 6.2. However, to simplify the sizing and synthesis process, this thesis builds the dynamic mission profile with only

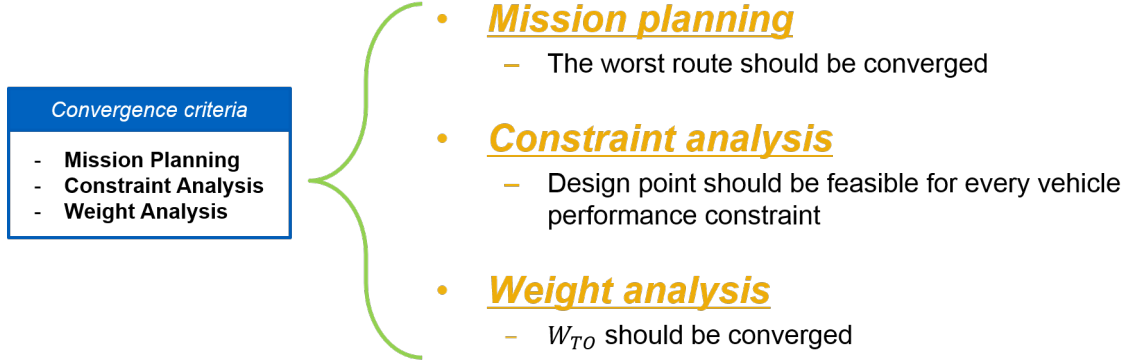


Figure 6.3: Convergence criteria of the FPI method for concurrent UAV design and vehicle routing problems

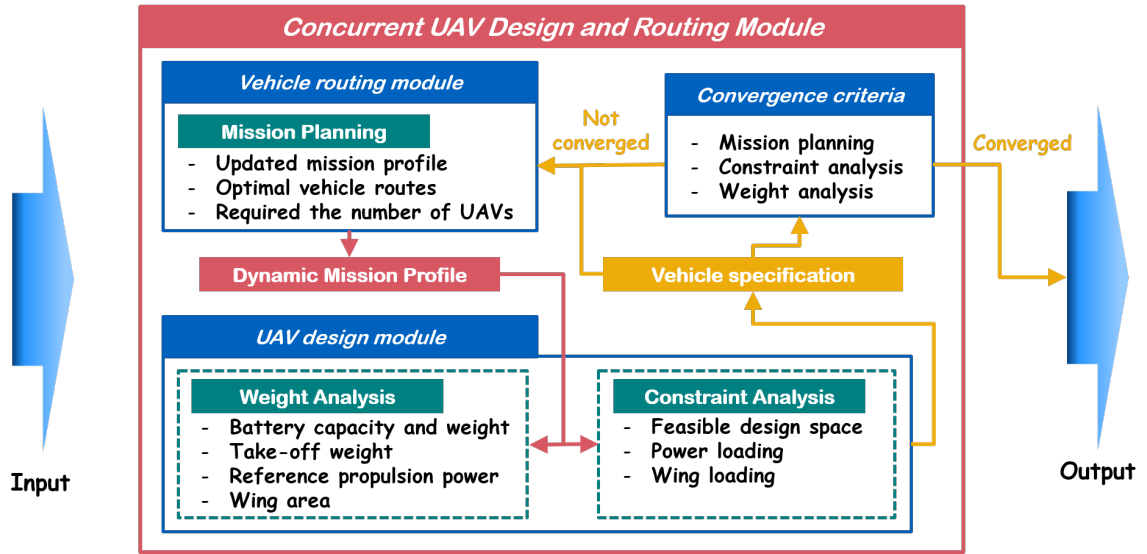


Figure 6.4: Structure of a FPI method for a concurrent UAV design and vehicle routing problem

takeoff/landing and cruise mission segments.

For the third part, three types of stop conditions can be considered as described in Figure 6.3. First, the mission profile from the vehicle routing module should be converged. This implies that the update of the UAV design module has no effect on the solution routes. Second, the vehicle design parameters should satisfy all performance constraints. This guarantees that the vehicle designed with the parameters can conduct the delivery mission planned by the vehicle routing module. Finally, the take-off weight of the UAV design module should be converged.

With these considerations about a FPI method, Figure 6.4 shows a structure of a FPI method to find a converged solution of a concurrent UAV design and vehicle routing problem. The outer loop of the integrated module is designed as a FPI. The UAV design module and the vehicle routing module work as subsystems in the integrated module.

The considerations about the structure of a FPI method inspires a hypothesis such that

Hypothesis for RQ 4 : By using the dynamic mission profile and the vehicle specification as coupled variables, a FPI method for concurrent UAV design and vehicle routing problems can be designed with convergence criteria in terms of mission planning, constraint analysis, and weight analysis.

The following section presents a framework for concurrent UAV design and routing problems which is designed by using the FPI structure presented in Hypothesis for RQ 4.

6.2 Creating a Framework for Concurrent UAV Design and Routing Problems for urban delivery

To address a concurrent UAV design and routing problem for urban delivery, the sizing and synthesis process presented in Chapter 5 are extended by adding the VRP optimization model formulated in Chapter 3 with an input graph described in Chapter 4. The developed framework for a concurrent UAV design and routing problem for urban delivery is illustrated in Figure 6.5.

The developed framework consists of five modules: urban flight network, endurance-constrained vehicle routing, UAV design, energy-constrained vehicle routing, and checking convergence modules. Note that this framework deals with each model as a black box function. Thus, if the interface between functions are matched, then any method, model, or framework can be integrated as a module into the presented framework.

First, the urban flight network module introduced in Chapter 4 creates collision-free paths among the depot and the delivery locations based on an urban model, then builds a flight network based on the paths as an input graph for optimization models.

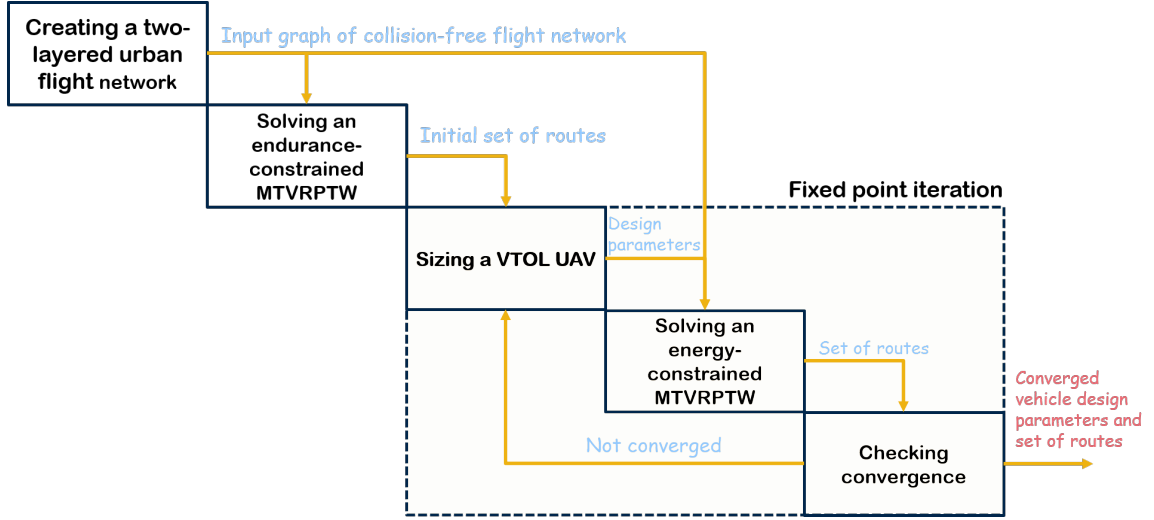


Figure 6.5: Flow chart of the integrated framework

Second, the endurance-constrained vehicle routing module presented in Chapter 3 builds solution routes for each vehicle based on both the input graph and initially assumed vehicle parameters. In the routes, the worst route is converted into a mission profile to represent the solution of the optimization model.

Third, the UAV design module described in Chapter 5 determines optimal vehicle design parameters such as P/W and W/S , then finds optimal physical parts to minimize take-off weight. To address more precise flight capacity of the designed vehicle, this thesis utilizes energy parameters [11, 132] such as

- Maximum battery capacity ($Energy_{max}$)
- The required power for forward flight (P_f)
- The required power for vertical flight (P_v)

Fourth, based on these energy parameters in the VRP module, an energy-constrained vehicle routing optimization model is presented, which can trace consumed energy by each vehicle. The optimization model is specifically described in the following subsection.

Lastly, the convergence of the FPI is evaluated in terms of mission planning, constraint analysis, and weight analysis as stated previously.

In this framework, by addressing a converged solution based on the FPI, the solution of the UAV design module has effects on the solution routes as well as the mission profile from the energy-constrained optimization model affects vehicle parameters. This is the key idea of the integrated framework for the concurrent UAV design and vehicle routing problems. In addition, this framework deals with each module as a black box function, thus, the various methods having the same interface can be independently integrated with this framework as mentioned previously.

The following subsection presents an Energy-constrained MTRVPTW optimization model to deal with energy concepts rather than endurance concepts to describe flight capacity of vehicles.

6.2.1 Optimization Model for an Energy-Constrained MTRVPTW

The presented optimization model for an energy-constrained MTRVPTW is an extension of that for an endurance-constrained MTRVPTW introduced in Chapter 3. To address energy concepts on UAV's operations, properties of UAV's propulsion system should be reflected into mathematical formulations to build total cost optimized routes. The energy constraints in formulations below are based on the formulations of Choi et al. [132] which is an extension of distance constraints of the previous works [130, 131, 133]. The energy constraints trace the energy used by each vehicle after departure from the depot. The energy constraints model a mission profile having take-off, cruise, and landing segments.

The objective function of the optimization model for an energy-constrained MTRVPTW can be written by

$$Min. \quad c_{fixed} \left(\sum_{k \in \mathcal{V}} \sum_{j \in \mathcal{C}} x_{0jk} - \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} \sigma_{ij} \right) + \sum_{k \in \mathcal{V}} \sum_{i \in \mathcal{N}_O} \sum_{j \in \mathcal{N}_D} c_{ijk} x_{ijk} \quad (6.1)$$

The objective function, Equation (6.1), minimizes the number of vehicles and total flight distance/hours. The constraints describes the ConOps of a UAS; each customer is visited

by a UAV exactly once, Equation (6.2).

$$\sum_{k \in \mathcal{V}} \sum_{j \in \mathcal{N}_D} x_{ijk} = 1, \quad (\forall i \in \mathcal{C}) \quad (6.2)$$

Each UAV can carry packages up to maximum payload weight, Equation (6.3).

$$\sum_{i \in \mathcal{C}} d_i \sum_{j \in \mathcal{N}_D} x_{ijk} \leq q, \quad (\forall k \in \mathcal{V}) \quad (6.3)$$

Each vehicle should depart/arrive at a depot, Equation (6.4) and (6.5), and it should leave the location after delivering a package, Equation (6.6).

$$\sum_{j \in \mathcal{N}_D} x_{0jk} = 1, \quad (\forall k \in \mathcal{V}) \quad (6.4)$$

$$\sum_{i \in \mathcal{N}_O} x_{i,n+1,k} = 1, \quad (\forall k \in \mathcal{V}) \quad (6.5)$$

$$\sum_{i \in \mathcal{N}_O} x_{ihk} - \sum_{j \in \mathcal{N}_D} x_{hjk} = 0, \quad (\forall h \in \mathcal{C}, \forall k \in \mathcal{V}) \quad (6.6)$$

When a vehicle visits more than two customers, the order of delivery time should correspond with the order of visits, Equation (6.7). This set of constraints provides subtour elimination.

$$s_{ik} + t_{ij} + \tau_i - s_{jk} \leq K(1 - x_{ijk}), \quad (\forall i \in \mathcal{N}_O, \forall j \in \mathcal{N}_D, \forall k \in \mathcal{V}) \quad (6.7)$$

The constraints for reusability of UAVs can be written by

$$\sum_{j \in \mathcal{C}} \sigma_{ij} - \sum_{k \in \mathcal{V}} x_{i(n+1)k} \leq 0, \quad (\forall i \in \mathcal{C}) \quad (6.8)$$

$$\sum_{i \in \mathcal{C}} \sigma_{ij} - \sum_{k \in \mathcal{V}} x_{0jk} \leq 0, \quad (\forall j \in \mathcal{C}) \quad (6.9)$$

$$\sum_{k \in \mathcal{V}} \sum_{j \in \mathcal{C}} x_{0jk} - \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}} \sigma_{ij} \leq M \quad (6.10)$$

$$s_{ik} + \tau_i + t_{i(n+1)} + \tau_{depot} + t_{0j} - s_{jk} \leq K(1 - \sigma_{ij}), \quad (\forall i, j \in \mathcal{C}, \forall k \in \mathcal{V}) \quad (6.11)$$

When there are more than two trips, the reuse of UAVs is considered, Equation (6.8) and (6.9). The total required number of vehicles is bounded by Equation (6.10). In order to reuse a UAV, the time enough to reload packages and swap batteries should be guaranteed, Equation (6.11).

To address energy consumption, a new set of design variables, e_{ijk} , is used, which trace the amount of used energy by each UAV to move from the depot to current location j . Each e_{ijk} is a real design variable. Then, the constraints of energy consumption can be written by

$$\sum_{j \in \mathcal{N}_D} e_{hjk} - \sum_{i \in \mathcal{N}_O} e_{ihk} = \sum_{j \in \mathcal{N}_D} (P_{to}t_{to} + P_ft_{hjk} + P_lt_l)x_{hjk}, (\forall h \in \mathcal{C}, \forall k \in \mathcal{V}) \quad (6.12)$$

$$e_{0jk} = (P_{to}t_{to} + P_ft_{0jk} + P_lt_l)x_{0jk}, (\forall j \in \mathcal{V}_D, \forall k \in \mathcal{V}) \quad (6.13)$$

$$e_{hjk} \geq (2P_{to} + P_f(t_{0hk} + t_{hjk}) + 2P_lt_l)x_{ihk}, (\forall h \in \mathcal{C}, \forall j \in \mathcal{V}_D, \forall k \in \mathcal{V}) \quad (6.14)$$

$$e_{ihk} \leq (Energy_{max,k} - P_{to}t_{to} - P_ft_{h(n+1)k} - P_lt_l)x_{ihk}, (\forall i \in \mathcal{V}_O, \forall h \in \mathcal{C}, \forall k \in \mathcal{V}) \quad (6.15)$$

$$e_{i(n+1)k} \leq Energy_{max,k}x_{i(n+1)k}, (\forall i \in \mathcal{V}_O, \forall k \in \mathcal{V}) \quad (6.16)$$

where P_{to} , P_f , and P_l are the required power for take-off, cruise, and landing respectively, t_{to} and t_l are take-off and landing time, t_{hjk} is the cruise time by a vehicle k from a node i to another node j , and $Energy_{max,k}$ is the maximum energy capacity of a vehicle k . Equation (6.12) traces energy consumption when a vehicle k moves from a node h to another node j . Equation (6.13) determines the initial energy consumption of each trip. Equation (6.14) and (6.15) provide lower and upper bounds of energy consumption at an intermediate node of each trip. Equation (6.16) guarantees that energy consumption of a vehicle k should be bounded by its maximum energy capacity. Other design variables are bounded by Equation

(6.17)-(6.19).

$$a_i \leq s_{ik} \leq b_i, \quad (\forall i \in \mathcal{C}, \forall k \in \mathcal{V}) \quad (6.17)$$

$$x_{ijk} \in \{0, 1\}, \quad (\forall i \in \mathcal{N}_O, \forall j \in \mathcal{N}_D, \forall k \in \mathcal{V}) \quad (6.18)$$

$$\sigma_{ij} \in \{0, 1\}, \quad (\forall i, j \in \mathcal{C}) \quad (6.19)$$

These energy-constrained formulations can be used to create trips for each UAVs after the power values are obtained from the vehicle sizing process introduced in Chapter 5.

This section presented a framework for concurrent UAV design and vehicle routing problems which utilizes a FPI method. To address Hypothesis for RQ 4, dynamic mission profiles and the vehicle specification are used as coupled variable between the UAV design module and the VRP module. Furthermore, the energy concept are integrated with the framework to more precisely describe flight capacity of vehicles. The following section executes experiments to demonstrate that the presented framework can address concurrent UAV design and vehicle routing problems by providing a converged solution of the UAV design module and the VRP module.

6.3 Experiments

This section designs and conducts experiments to demonstrate the hypothesis such that

Hypothesis for RQ 4 : By using the dynamic mission profile and the vehicle specification as coupled variables, a FPI method for concurrent UAV design and vehicle routing problems can be designed with convergence criteria in terms of mission planning, constraint analysis, and weight analysis.

To address the hypothesis, a framework for concurrent UAV design and vehicle routing problems for urban delivery is presented in the previous section. To substantiate the hypothesis with the presented framework, this section conducts two experiments such that

Experiment 7 : To evaluate the convergence property of the FPI method addressing the concurrent UAV design and vehicle routing problems, this experiment executes convergence analysis.

Experiment 8 : To address the best vehicle's operating speed and maximum payload capacity for given scenarios, this experiment executes sensitivity analysis with various conditions of vehicle's operating speed and maximum payload capacity.

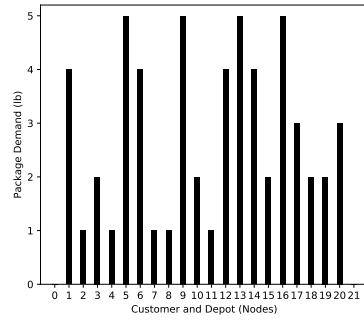
The fundamental intent of these experiments is to conduct integration tests of the framework including the entire modules presented in this thesis. The framework is mainly based on a FPI method including the UAV design module and the VRP module. Thus, the two experiments focus on analyzing the FPI method with convergence analysis and sensitivity analysis.

Specifically, Experiment 7 conducts convergence analysis with diverse initial values of the FPI with given scenarios. The initial value of the FPI is the worst-case route of the endurance-constrained MTRPTW presented in Chapter 3 that depends on the maximum endurance of vehicles. Thus, Experiment 7 executes convergence analysis of the developed framework with various maximum endurance values. Experiment 8 conducts sensitivity analysis in terms of vehicle's operating speed and maximum payload capacity to find the best option for given scenarios. Note that as an effort to address a better solution, Experiment 8 executes each simulation with various initial values of the FPI.

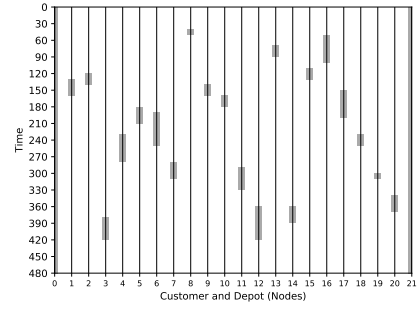
For Experiment 7 and 8, there are common assumptions. The assumptions for the UAV design module described in 5.3 are also used for the two experiments. In the VRP module, a simple vehicle cost model is assumed to represent changing vehicle cost by the result of the UAV design module, which is defined by

$$C_{fixed} = (1000 + W_{payload} * 100) * \frac{W_{to}}{55} \quad (6.20)$$

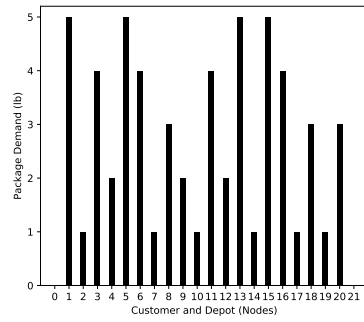
In this cost model, vehicle cost is represented by UAV's payload capacity and take-off



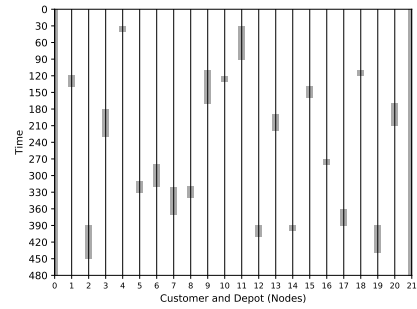
(a) First scenario: distribution of package demand



(b) First scenario: distribution of time windows



(c) Second scenario: distribution of package demand



(d) Second scenario: distribution of time windows

Figure 6.6: The two demand scenario

weight. Since take-off weight is changed for every UAV sizing iteration, the energy-constrained MTRPTW optimization model presented in Section 6.2.1 uses the updated vehicle cost when finding solution routes for each iteration of the FPI.

Service time for each delivery is assumed as 3 minutes to address times for landing, margin, and take-off. The maximum payload weight of UAVs is assumed as 5 lb. Two package demand scenarios are stochastically created based on a uniform distribution as illustrated in Figure 6.6. The distribution of package demand of two scenarios as illustrated in Figure 6.6a and 6.6c are similar each other while the second scenario has tighter time windows visualized in Figure 6.6d than that of the first scenario as shown in Figure 6.6b. The demand scenario with looser time windows has more freedom to plan a vehicle schedule. For the other conditions, the assumptions of experiments in Section 4.3 are used with

the San Diego urban model presented in Chapter 4.

6.3.1 Experiment 7: Convergence Analysis

The convergence of the FPI methods depends on initial values. The converged value is changed by initial value. To address a solution closed to a global minimum in an MDO problem, a set of initial values can be used rather than a single initial value. Thus, this experiment executes convergence analysis of the developed framework for concurrent UAV design and routing problems with various initial values. The initial value of the framework is the mission profile of the worst-case route of the endurance-constrained MTRVPTW optimization model presented in Chapter 3. The worst-case route depends on the maximum endurance of UAVs. Thus, the initial value of the developed framework also is the maximum endurance of UAVs. In the San Diego urban model presented in Chapter 4, the farthest delivery location from the depot requires at least 30-minute endurance without service time. Thus, this experiment explores the solution space with the initial values of 35- ~ 60-minute endurance.

The result of experiments for evaluating effects of initial conditions are summarized in Table 6.1 ~ 6.3, where VRP column indicates the results of the endurance-constrained MTRVPTW optimization model, and MDO column shows the results of the framework for concurrent UAV design and routing, and difference is calculated as

$$Diff(\%) = \frac{(MDO - VRP)}{VRP} \times 100 \quad (6.21)$$

Note that the values in VRP columns of vehicle cost, objective function, and W_{to} are modified by a post-processing. Specifically, after solving the endurance-constrained MTRVPTW model, a UAV sizing process is conducted once with a mission profile created by the worst route of the optimization model. Using this post-processing, the weight of UAVs of the model can be calculated. Then, the vehicle cost can be calculated by Equation

Table 6.1: Summary of initial value analysis of the objective function

Demand Scen.		Obj. function		
		VRP	MDO	Diff (%)
First	Min.	1839.54	1790.07	-2.69
	Avg.	1894.98	1806.01	-4.70
	Stdev.	74.90	7.65	-89.78
Second	Min.	1806.51	1817.87	0.63
	Avg.	1962.44	1921.55	-2.08
	Stdev.	100.69	80.14	-20.41

Table 6.2: Summary of initial value analysis of the flight time

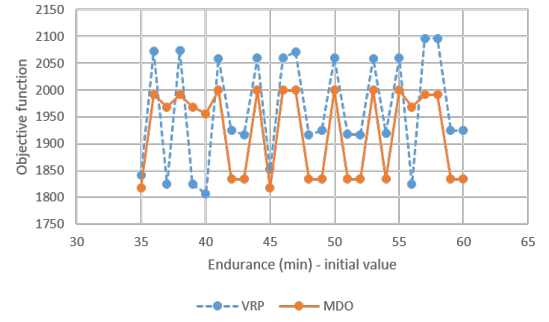
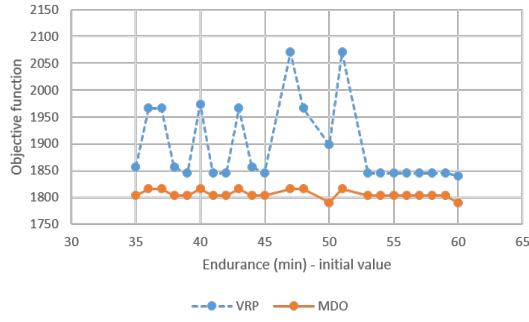
Demand Scen.		Flight time (min)		
		VRP	MDO	Diff (%)
First	Min.	234.58	210.11	-10.43
	Avg.	237.50	214.90	-9.51
	Stdev.	4.98	6.12	22.94
Second	Min.	223.02	223.33	0.14
	Avg.	228.69	227.66	-0.45
	Stdev.	5.61	5.07	-9.57

Table 6.3: Summary of initial value analysis of the vehicle cost

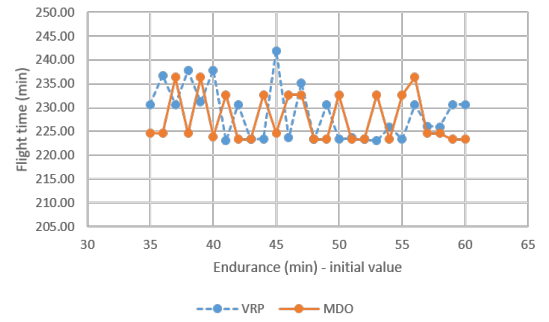
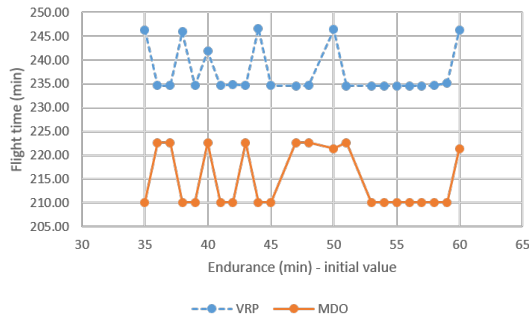
Demand Scen.		Vehicle cost		
		VRP	MDO	Diff (%)
First	Min.	796.62	784.36	-1.54
	Avg.	828.74	795.55	-4.00
	Stdev.	37.96	3.53	-90.70
Second	Min.	784.36	796.62	1.56
	Avg.	866.87	846.94	-2.30
	Stdev.	51.30	38.36	-25.22

6.20, and objective function also can be calculated. These values are calculated to compare both approaches: the sequential approach and the concurrent approach to solve UAV design and routing problems.

All experiment results of convergence analysis are specified in Table 6.4 and 6.5 that

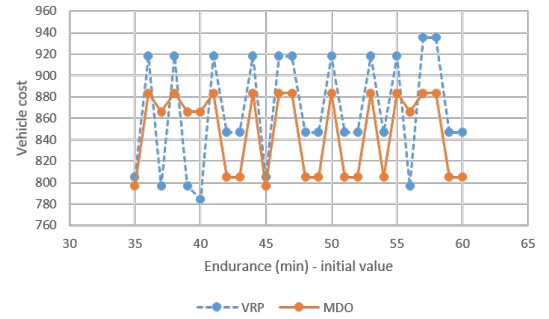
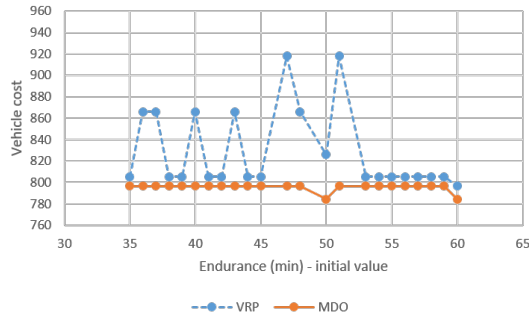


(a) First scenario: convergence of objective function (b) Second scenario: convergence of objective function



(c) First scenario: convergence of flight time

(d) Second scenario: convergence of flight time



(e) First scenario: convergence of vehicle cost

(f) Second scenario: convergence of vehicle cost

Figure 6.7: Results of convergence analysis of two demand scenarios

are attached at the end of this subsection. Figure 6.7 visualizes the experiment results; the plots shows objective function, flight time, and vehicle cost in terms of initial value which is maximum endurance of vehicles for endurance-constrained MTVRPTW model. In the first demand scenario, the objective function value in Figure 6.7a and Table 6.1 decreases when using the concurrent framework. The decrease happens in both flight time in Figure 6.7c and Table 6.2 and vehicle cost in Figure 6.7e and Table 6.3. The result show that the energy-

constrained MTVRPTW optimization model has a tendency to find solutions having less total flight time than the endurance-constrained MTVRPTW optimization model as shown in 6.7c. This tendency is expected because it can happen that a UAV has energy enough to deliver a package to another location after its maximum endurance time. Moreover, Figure 6.7e describes that the sizing process designs a vehicle having the minimum take-off weight for a given mission profile during MDO iterations. Note that the variation of MDO iterations happens because of the sizing process is a component-based method; the design values are discretely varied by selecting physical components. For instance, in the sizing process, when a battery is designed, its capacity is larger than or equal to the required amount of energy of a given mission profile. If the exceeded battery capacity is enough to change the solution of the energy-based MTVRPTW optimization model, it makes the next MDO iteration. For some initial endurance values, the energy-constraint MTVRPTW model returns no feasible solution. This result happens because the lower bound of the energy-constrained MTVRPTW model is higher than that of the endurance-constrained MTVRPTW model. This leads to a better solution in a safety point of view.

In the second scenario, the minimum and average of the objective function value is relatively similar to each other as illustrated in Figure 6.7. This small difference happens because of the component-based sizing process; flight time of two approaches are similar each other as described in Figure 6.7d while vehicle cost of the VRP line is slightly higher as shown in Figure 6.7f. However, the results of the second scenario in Table 6.1 ~ 6.3 show that the concurrent framework is more robust to initial value's variation; The standard deviation of solutions from the concurrent framework is less than that from the endurance-constrained MTVRPTW optimization model. This is another benefit of the developed concurrent framework.

The results of convergence analysis with two scenarios show that the FPI in the developed framework provides converged solutions with various initial values. The results also describe that the developed framework provides better solutions in terms of the total cost

or the robustness to changing initial values than the endurance-constrained MTVRPTW optimization model for given scenarios. Thus, these results substantiate that the developed framework provides a converged solution which can provide the lower total cost or the robustness to initial values.

Table 6.4: Effect of initial conditions on the solution of both the endurance-constrained MTPRTW and the framework for concurrent vehicle design and routing with the first demand scenario

Endu- rance	Vehicle cost			Obj. function			Total flight time (min)			# of req. UAVs		W_{to} (lb)		
	VRP	MDO	Diff (%)	VRP	MDO	Diff (%)	VRP	MDO	Diff (%)	VRP	MDO	VRP	MDO	Diff (%)
35	805.17	796.62	-1.06	1856.67	1803.35	-2.87	246.33	210.11	-14.7	2	2	29.52	29.21	-1.06
36	866	796.62	-8.01	1966.61	1815.88	-7.66	234.62	222.64	-5.11	2	2	31.75	29.21	-8.01
37	866	796.62	-8.01	1966.61	1815.88	-7.66	234.62	222.64	-5.11	2	2	31.75	29.21	-8.01
38	805.17	796.62	-1.06	1856.24	1803.35	-2.85	245.90	210.11	-14.56	2	2	29.52	29.21	-1.06
39	805.17	796.62	-1.06	1844.96	1803.35	-2.25	234.62	210.11	-10.45	2	2	29.52	29.21	-1.06
40	866	796.62	-8.01	1973.88	1815.88	-8	241.88	222.64	-7.96	2	2	31.75	29.21	-8.01
41	805.17	796.62	-1.06	1844.96	1803.35	-2.25	234.62	210.11	-10.45	2	2	29.52	29.21	-1.06
42	805.17	796.62	-1.06	1845.2	1803.35	-2.27	234.86	210.11	-10.54	2	2	29.52	29.21	-1.06
43	866	796.62	-8.01	1966.61	1815.88	-7.66	234.62	222.64	-5.11	2	2	31.75	29.21	-8.01
44	805.17	796.62	-1.06	1856.95	1803.35	-2.89	246.61	210.11	-14.8	2	2	29.52	29.21	-1.06
45	805.17	796.62	-1.06	1844.96	1803.35	-2.25	234.62	210.11	-10.45	2	2	29.52	29.21	-1.06
47	918.03	796.62	-13.22	2070.64	1815.88	-12.3	234.58	222.64	-5.09	2	2	33.66	29.21	-13.22
48	866	796.62	-8.01	1966.61	1815.88	-7.66	234.62	222.64	-5.11	2	2	31.75	29.21	-8.01
50	825.98	784.36	-5.04	1898.33	1790.07	-5.7	246.37	221.36	-10.15	2	2	30.29	28.76	-5.04
51	918.03	796.62	-13.22	2070.64	1815.88	-12.3	234.58	222.64	-5.09	2	2	33.66	29.21	-13.22
53	805.17	796.62	-1.06	1844.92	1803.35	-2.25	234.58	210.11	-10.43	2	2	29.52	29.21	-1.06
54	805.17	796.62	-1.06	1844.92	1803.35	-2.25	234.58	210.11	-10.43	2	2	29.52	29.21	-1.06
55	805.17	796.62	-1.06	1844.92	1803.35	-2.25	234.58	210.11	-10.43	2	2	29.52	29.21	-1.06
56	805.17	796.62	-1.06	1844.92	1803.35	-2.25	234.58	210.11	-10.43	2	2	29.52	29.21	-1.06
57	805.17	796.62	-1.06	1844.92	1803.35	-2.25	234.58	210.11	-10.43	2	2	29.52	29.21	-1.06
58	805.17	796.62	-1.06	1844.96	1803.35	-2.25	234.62	210.11	-10.45	2	2	29.52	29.21	-1.06
59	805.17	796.62	-1.06	1845.56	1803.35	-2.29	235.21	210.11	-10.68	2	2	29.52	29.21	-1.06
60	796.62	784.36	-1.54	1839.54	1790.07	-2.69	246.30	221.36	-10.12	2	2	29.21	28.76	-1.54

Table 6.5: Effect of initial conditions on the solution of both the endurance-constrained MTRPTW and the framework for concurrent vehicle design and routing with the second demand scenario

Endu- rance	Vehicle cost				Obj. function				Total flight time (min)				# of req. UAV's		W_{to} (lb)			
	VRP	MDO	Diff (%)		VRP	MDO	Diff (%)		VRP	MDO	Diff (%)		VRP	MDO		VRP	MDO	Diff (%)
35	805.17	796.62	-1.06		1840.94	1817.87	-1.25		230.6	224.63	-2.59		2	2		29.52	29.21	-1.06
36	918.03	883.34	-3.78		2072.81	1991.24	-3.94		236.75	224.56	-5.15		2	2		33.66	32.39	-3.78
37	796.62	866	8.71		1823.89	1968.37	7.92		230.64	236.37	2.48		2	2		29.21	31.75	8.71
38	918.03	883.34	-3.78		2073.77	1991.24	-3.98		237.71	224.56	-5.53		2	2		33.66	32.39	-3.78
39	796.62	866	8.71		1824.38	1968.37	7.89		231.13	236.37	2.27		2	2		29.21	31.75	8.71
40	784.36	866	10.41		1806.51	1955.82	8.27		237.79	223.83	-5.87		2	2		28.76	31.75	10.41
41	918.03	883.34	-3.78		2059.08	1999.36	-2.9		223.02	232.67	4.33		2	2		33.66	32.39	-3.78
42	846.79	805.17	-4.92		1924.19	1833.67	-4.7		230.6	223.33	-3.15		2	2		31.05	29.52	-4.92
43	846.79	805.17	-4.92		1916.92	1833.67	-4.34		223.33	223.33	0		2	2		31.05	29.52	-4.92
44	918.03	883.34	-3.78		2059.44	1999.36	-2.92		223.38	232.67	4.16		2	2		33.66	32.39	-3.78
45	805.17	796.62	-1.06		1852.24	1817.87	-1.86		241.91	224.63	-7.14		2	2		29.52	29.21	-1.06
46	918.03	883.34	-3.78		2059.71	1999.36	-2.93		223.65	232.67	4.03		2	2		33.66	32.39	-3.78
47	918.03	883.34	-3.78		2071.24	1999.36	-3.47		235.18	232.67	-1.07		2	2		33.66	32.39	-3.78
48	846.79	805.17	-4.92		1916.92	1833.67	-4.34		223.33	223.33	0		2	2		31.05	29.52	-4.92
49	846.79	805.17	-4.92		1924.19	1833.67	-4.7		230.6	223.33	-3.15		2	2		31.05	29.52	-4.92
50	918.03	883.34	-3.78		2059.39	1999.36	-2.92		223.33	232.67	4.18		2	2		33.66	32.39	-3.78
51	846.79	805.17	-4.92		1917.24	1833.67	-4.36		223.65	223.33	-0.14		2	2		31.05	29.52	-4.92
52	846.79	805.17	-4.92		1916.92	1833.67	-4.34		223.33	223.33	0		2	2		31.05	29.52	-4.92
53	918.03	883.34	-3.78		2059.08	1999.36	-2.9		223.02	232.67	4.33		2	2		33.66	32.39	-3.78
54	846.79	805.17	-4.92		1919.48	1833.67	-4.47		225.9	223.33	-1.14		2	2		31.05	29.52	-4.92
55	918.03	883.34	-3.78		2059.39	1999.36	-2.92		223.33	232.67	4.18		2	2		33.66	32.39	-3.78
56	796.62	866	8.71		1823.84	1968.37	7.92		230.6	236.37	2.5		2	2		29.21	31.75	8.71
57	935.37	883.34	-5.56		2096.82	1991.24	-5.04		226.08	224.56	-0.67		2	2		34.3	32.39	-5.56
58	935.37	883.34	-5.56		2096.64	1991.24	-5.03		225.9	224.56	-0.59		2	2		34.3	32.39	-5.56
59	846.79	805.17	-4.92		1924.19	1833.67	-4.7		230.6	223.33	-3.15		2	2		31.05	29.52	-4.92
60	846.79	805.17	-4.92		1924.19	1833.67	-4.7		230.6	223.33	-3.15		2	2		31.05	29.52	-4.92

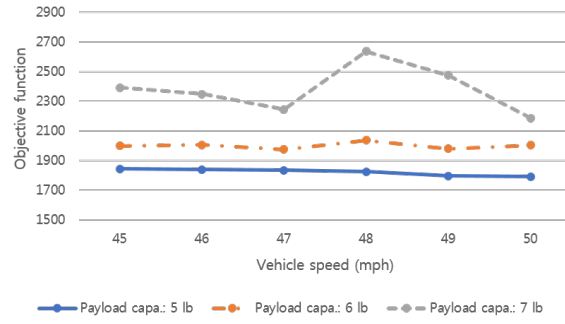
6.3.2 Experiment 8: Sensitivity Analysis

This experiment conducts a sensitivity analysis of the developed framework for concurrent UAV design and routing in terms of vehicle payload capacity and vehicle speed; these two parameters are commonly used as system-level inputs in [30, 31, 32]. The variations on objective function, total flight time, vehicle cost, and the number of UAVs are observed. In this experiment, the same demand scenarios are used. Furthermore, the vehicle speed range from 45 to 50 mph and the payload capacity range from 5 to 7 lb are explored.

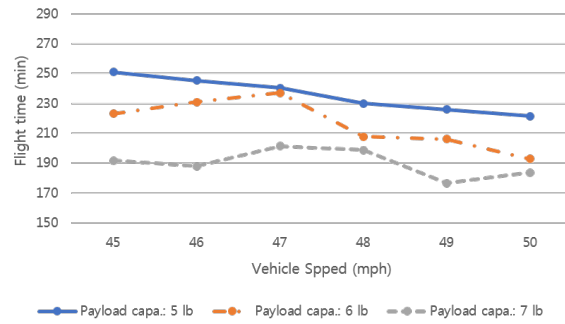
The results of the first demand scenario are shown in Table 6.6. Note that each row represents the best result of the concurrent framework with 35-min \sim 60-min endurance as initial values like the previous experiment. The results are visualized in Figure 6.8. At first, flight time decreases as either vehicle speed or payload capacity increases as shown in Figure 6.8b. This result makes sense because although all routes are the same, if vehicle speed increases, then flight time decreases. Furthermore, the vehicle having larger payload capacity has more opportunity to deliver more packages on a trip. This leads to the shorter total flight time. Second, as shown in Figure 6.8c, vehicle cost depends on payload capacity and take-off weight like Equation 6.20, and take-off weight is determined by the component selected by the sizing module. In this sense, the vehicle cost of the vehicle having 7-lb payload capacity and operating at 48 mph increases sharply. This is because the vehicle requires the heavier propulsion system. This happens because of the component-based sizing method which provide non-linear results inherently. Next, Figure 6.8a shows the same curve since it is determined by vehicle cost, flight time, and the number of vehicles. Finally, there is no change on the number of UAVs required to deliver all packages as shown in Figure 6.8.

Table 6.6: Summary of sensitivity analysis of the first demand scenario

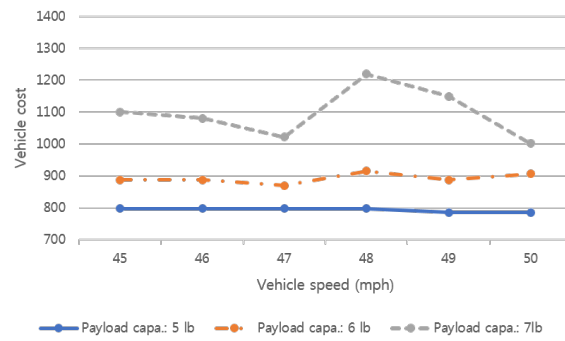
Vehicle speed (mph)	Payload capa. (lb)	Obj. func.	Flight time (min)	Vehicle cost	# of UAVs
45	5	1844.26	251.02	796.62	2
	6	1997.56	223.22	887.17	2
	7	2392.4	191.71	1100.35	2
46	5	1838.56	245.32	796.62	2
	6	2006.06	231.01	887.17	2
	7	2349.21	187.84	1080.69	2
47	5	1833.58	240.34	796.62	2
	6	1974.3	236.95	868.67	2
	7	2244.85	201.41	1021.72	2
48	5	1823.23	229.99	796.62	2
	6	2037.4	207.82	914.79	2
	7	2638.88	198.66	1220.11	2
49	5	1794.59	225.88	784.36	2
	6	1980.41	206.06	887.17	2
	7	2475.31	176.62	1149.35	2
50	5	1790.07	221.36	784.36	2
	6	2004.22	192.87	905.67	2
	7	2187.8	183.67	1002.06	2



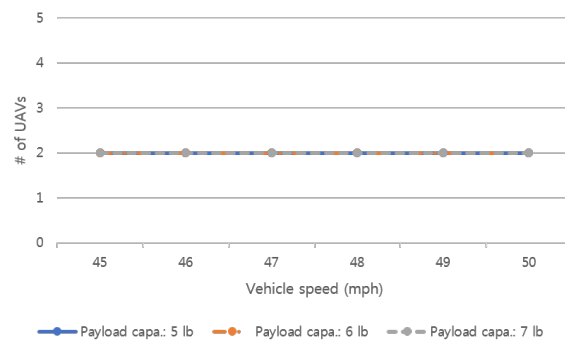
(a) Objective function



(b) Flight time



(c) Vehicle cost



(d) The number of UAVs

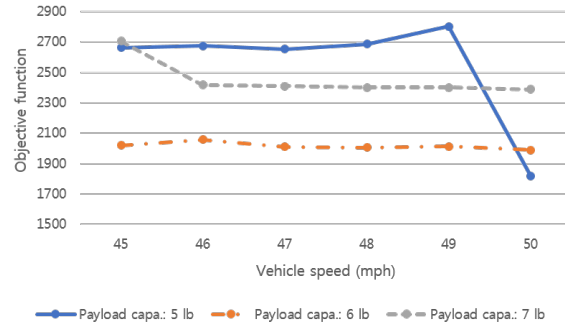
Figure 6.8: Results of sensitivity analysis of the first demand scenario

The results of the second demand scenario are shown in Table 6.7. The results are visualized in Figure 6.9. At first, the overall effects of vehicle speed and payload capacity on flight time has the same tendency with the first scenario as illustrated in Figure 6.9b; flight time is decreased as either vehicle speed or payload capacity increases. However, a vehicle having 5-lb payload capacity and operating at 50 mph seems to have a different tendency. This result happens because of both the change of the number of required UAVs at the condition as shown in Figure 6.8d and time window constraints. Specifically, as the number of UAVs decreases, the opportunity to visit another delivery location within its time window also decreases. Since this behavior leads to the more number of reloading, the total flight time could increase. However, by increasing the number of reloading, UAVs need not to have large capacity batteries. This decreases the vehicle cost as shown in Figure 6.9c. Furthermore, since the number of required UAVs is decreased, the objective function also decreases as illustrated in Figure 6.9a. Thus, this sensitivity analysis result also is reasonable.

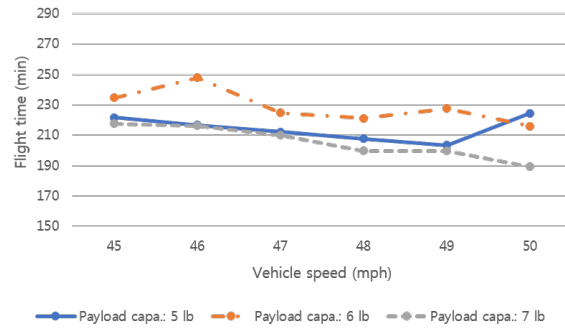
These sensitivity analyses show that payload capacity has larger effects on the results than vehicle speed for the scenarios. However, the vehicle speed also can have significant effects on the result like the case for the vehicle having 5-lb payload capacity and operating at 50 mph with the second demand scenario. Furthermore, the result can be used to find the best values of the parameters for given scenarios. In this case, the condition of both 5-lb payload capacity and 50-mph vehicle speed is the best option for both scenarios, but the second best option for each scenario is different; the condition of 5-lb payload capacity and 49-mph vehicle speed is the second best for the first scenario whereas the condition of 6-lb payload capacity and 50-mph vehicle speed is the second best for the second scenario. These results demonstrate that the developed framework for concurrent UAV design and routing problems for urban delivery works well for the given scenarios; additionally, these results show how trade studies can be executed to determine vehicle's operating speed and payload capacity by utilizing delivery scenarios.

Table 6.7: Summary of sensitivity analysis of the second demand scenario

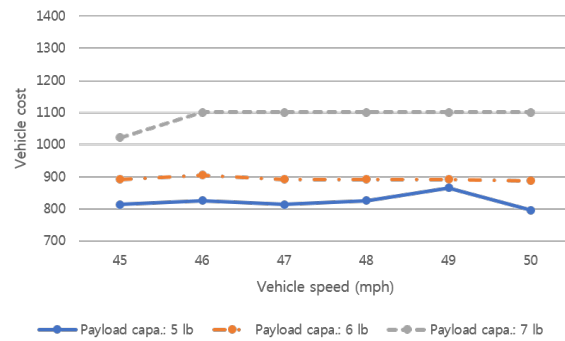
Vehicle speed (mph)	Payload capa. (lb)	Obj. func.	Flight time (min)	Vehicle cost	# of UAVs
45	5	2663.45	221.56	813.97	3
	6	2019.89	234.71	892.59	2
	7	2705.12	217.73	1021.72	2
46	5	2674.69	216.74	825.98	3
	6	2059.34	247.99	905.67	2
	7	2416.87	216.17	1100.35	2
47	5	2654.02	212.13	813.97	3
	6	2009.86	224.68	892.59	2
	7	2410.87	210.18	1100.35	2
48	5	2685.65	207.71	825.98	3
	6	2006.27	221.09	892.59	2
	7	2400.35	199.66	1100.35	2
49	5	2801.46	203.47	866	3
	6	2012.83	227.65	892.59	2
	7	2400.35	199.66	1000.35	2
50	5	1817.87	224.63	796.62	2
	6	1990.29	215.94	887.17	2
	7	2389.99	189.3	1100.35	2



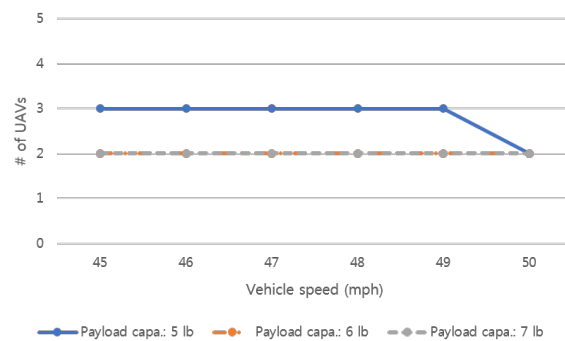
(a) Objective function



(b) Flight time



(c) Vehicle cost



(d) The number of UAVs

Figure 6.9: Results of sensitivity analysis of the first second scenario

6.4 Conclusions

This chapter initiated from the research question such that

Research Question 4 : How can the small VTOL UAV design module and the VRP module be integrated with a FPI method to obtain a converged solution?

To address this research question, the structure and interface of the UAV design module and the VRP module were analyzed. From the analysis, there was an observation that the dynamic mission profile and the vehicle specification can be worked as coupled variables between the UAV design module and the VRP module. The convergence criteria also are required for the FPI method, which consists of three types: mission planning, constraint analysis, and weight analysis. These results inspired a hypothesis such that

Hypothesis for RQ 4 : By using the dynamic mission profile and the vehicle specification as coupled variables, a FPI method for concurrent UAV design and vehicle routing problems can be designed with convergence criteria in terms of mission planning, constraint analysis, and weight analysis.

To address Hypothesis for RQ 4, a framework for concurrent UAV design and routing problems for urban delivery is developed based on a FPI method to find a converged solution of both the UAV design module and the VRP module. The developed framework consists of five modules: urban flight network, endurance-constrained vehicle routing, UAV design, energy-constrained vehicle routing, and checking convergence modules. The urban flight network module provides urban flight network to address collision-free paths. the endurance-constrained vehicle routing module provides an initial value of the FPI method. The UAV design, the energy-constrained vehicle routing, and the checking convergence modules compose a FPI to find a converged solution.

To prove Hypothesis for RQ 4, Experiment 7 and 8 were executed as convergence analysis and sensitivity analysis respectively. Experiment 7 revealed that the developed

framework has a potential of decreasing the objective function value by reducing either total flight time or vehicle cost for the given scenarios. Furthermore, the developed framework produced a solution with less deviations in terms of initial values. Experiment 8 showed that sensitivity analysis using the developed framework can lead to a better decision on design parameters for a UAS-based delivery system and determine which parameter has a larger effect on a solution. These results show that the developed framework is effective to both build solutions and analyze them for a UAS-based delivery system. Therefore, the results lead to a conclusion that Hypothesis for RQ 4 is demonstrated.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This thesis initiated by examining an aircraft sizing and synthesis process to address sUAS-based urban delivery systems. This examination showed that for delivery missions, an aircraft sizing and synthesis process for a fleet-level mission should be used to address total cost optimized mission planning and vehicle design in terms of the total cost of the delivery system.

In order to further examine the observation, a literature review on concurrent aircraft vehicle design and routing problems was conducted. The literature review revealed that existing methods do not find a converged solution of aircraft vehicle design module and vehicle routing module while the concurrent problem is a coupled problem of the two modules. This observation motivated this thesis to research and develop a framework for concurrent UAV design and routing for urban delivery systems which extends the existing methods in terms of

- Allowing to design a small VTOL UAV in a UAV design module
- Formulating a VRP for UAS-based delivery systems
- Including an urban model to address collision-free paths
- Finding a solution of a coupled problem of the small VTOL UAV design module and the VRP module for urban delivery

These extensions are directly associated with research questions in this thesis such that

Research Question 1 (RQ 1) : How can a set of main components of small fixed-wing VTOL UAVs as a result of a sizing and synthesis process be obtained?

Research Question 2 (RQ 2) : What type of a VRP optimization model best represents the operations of sUAS-based delivery systems?

Research Question 3 (RQ 3) : What type of an input graph of a VRP model best represents collision-free paths in the urban environment for sUAS-based delivery systems?

Research Question 4 (RQ 4) : How can the small VTOL UAV design module and the VRP module be integrated with a FPI method to obtain a converged solution?

Furthermore, these research questions inspired this thesis's research objective such that

Research Objective (RO) : To develop a novel sizing and synthesis approach for small VTOL UAV that takes into account both planning vehicle operation on the flight network in which it will be operating and designing the flight network capable of addressing the obstructed environment as part of the vehicle design process.

At first, to address Research Question 2 (RQ 2), a literature review on the Vehicle Routing Problem (VRP) relevant to sUAS-based delivery systems was conducted and observed that the Multi-Trip Vehicle Routing Problem with Time Windows (MTVRPTW) optimization model can address on-demand delivery systems with the concept of reusing vehicles. However, the optimization model cannot address the property of vehicles having limited endurance. The observation inspired the hypothesis such that

Hypothesis for RQ 2 : To address operations of sUAS-based delivery systems, an endurance-constrained MTVRPTW model should be utilized.

To prove this hypothesis, an endurance-constrained MTVRPTW model was presented in Chapter 3, and two experiments were conducted such that

Experiment 1 : To find what type of VRP model best represents a UAS package delivery network, this experiment compares three models: VRPTW, endurance-constrained VRPTW, and endurance-constrained MTVRPTW models.

Experiment 2 : To address a tendency of solutions by reusability and maximum endurance conditions, this experiment executes sensitivity analysis of the constraints with two models: VRPTW and endurance-constrained MTVRPTW models.

Experiment 1 showed that the presented model can address the properties of both reuse of vehicles and limited-endurance vehicles for on-demand delivery systems. Experiment 2 executed a sensitivity analysis of maximum endurance of vehicles and demonstrated that the endurance-constrained MTVRPTW model should be utilized for a delivery system using short endurance vehicles to obtain a total cost optimized schedule which minimizes total cost. These results lead to a conclusion that Hypothesis for RQ 2 is confirmed.

Second, to deal with Research Question 3 (RQ 3), a literature review on path-planning method to address collision-free paths was executed. The observation from the literature review was that road map methods can efficiently represent the obstructed environment as an input graph of a VRP optimization model. Furthermore, the Voronoi diagram which is a road map method can efficiently address a large-scale obstructed area such as the urban environment while it provides the safest paths. This observation inspired that hypothesis such that

Hypothesis for RQ 3 : To address collision-free paths as an input graph of a VRP for the urban environment, a Voronoi-diagram-based method should be utilized.

For this hypothesis, a framework for a two-layered urban flight network was presented. The framework consists of four steps: creating an urban model, capturing urban obstacles and restricted airspace, creating a low-level urban flight network based on a Voronoi diagram, and creating a high-level urban flight network to transfer a Voronoi diagram into an input graph of a VRP optimization model. To prove Hypothesis for RQ 3, two experiments were executed such that

Experiment 3 : To find which type of flight networks better represents collision-free paths in the urban environment, this experiment compares the solution routes of a point-to-

point flight network with that of a two-layered urban flight network.

Experiment 4 : To address the best vehicle specification for a given scenario, this experiment conducts sensitivity analysis of maximum endurance and maximum payload capacity constraints with the integrated module with both an endurance-constrained MTRPTW model and a two-layered urban flight network.

Experiment 3 demonstrated that to address collision-free paths in an obstructed urban area, the two-layered urban flight network built by the framework presented in Chapter 4 should be utilized with a VRP optimization model. Experiment 4 which is sensitivity analysis also demonstrated that the endurance-constrained MTRPTW model works well with the two-layered urban flight network for the given scenarios. These results lead to a conclusion that Hypothesis for RQ 3 is substantiated.

Third, to address Research Question 1 (RQ 1), a literature review on sizing and synthesis process for sVTOL UAVs was conducted. The observation from the literature review was that for small fixed-wing VTOL UAVs, non-parametric sizing and synthesis processes mainly addressing the propulsion system should be utilized. This observation inspired the hypothesis such that

Hypothesis for RQ 1 : To obtain a set of components of a propulsion system for small fixed-wing VTOL UAVs, a component-based sizing and synthesis process can be utilized.

To prove Hypothesis for RQ 1, a component-based sizing and synthesis process for small fixed-wing VTOL UAVs was implemented, which can provide a set of components based on component database. To validate the implemented process, two experiments were executed such that

Experiment 5 : This experiment conducts a sizing and synthesis process for a single-stop delivery mission to validate the implemented sizing and synthesis process.

Experiment 6 : This experiment executes a sizing and synthesis process for a two-stop delivery mission to validate the implemented sizing and synthesis process.

Experiment 5 and 6 were conducted with two delivery scenarios: a single-/two-stop delivery mission respectively. The former scenario represents a single-package delivery UAV while the latter scenario describes a multi-package delivery UAV. The results of two experiments demonstrated the fact that the implemented sizing and synthesis process for small fixed-wing VTOL UAVs provides feasible and reasonable design parameters of P/W , W/S , and take-off weight. The results lead to a conclusion that Hypothesis for RQ 1 is substantiated.

Finally, to deal with Research Question 4 (RQ 4), the structure and interface of the UAV design module and the VRP module were analyzed. The analysis led to an observation that to address a FPI method, the dynamic mission profile and the vehicle specification can be utilized as coupled variables between the UAV design module and the VRP module with convergence criteria in terms of mission planning, constraint analysis, and weight analysis. These results inspire a hypothesis such that

Hypothesis for RQ 4 : By using the dynamic mission profile and the vehicle specification as coupled variables, a FPI method for concurrent UAV design and vehicle routing problems can be designed with convergence criteria in terms of mission planning, constraint analysis, and weight analysis.

For the hypothesis, a framework for concurrent UAV design and vehicle routing problems for urban delivery was developed. The developed framework consists of five modules: urban flight network, endurance-constrained vehicle routing, UAV design, energy-constrained vehicle routing, and checking convergence modules. The UAV design, the energy-constrained vehicle routing, and the checking convergence modules compose a FPI to find a converged solution. To address Hypothesis for RQ 4, two experiments were conducted such that

Experiment 7 : To evaluate the convergence property of the FPI method addressing the concurrent UAV design and vehicle routing problems, this experiment executes convergence analysis.

Experiment 8 : To address the best vehicle's operating speed and maximum payload capacity for given scenarios, this experiment executes sensitivity analysis with various conditions of vehicle's operating speed and maximum payload capacity.

Experiment 7 revealed that the developed framework has a potential of decreasing the objective function value by reducing either total flight time or vehicle cost for the given scenarios. Moreover, the developed framework produced a solution with less deviations in terms of initial values. Experiment 8 showed that sensitivity analysis using the developed framework can lead to a better decision on design parameters for a UAS-based delivery system and determine which parameter has a larger effect on a solution. These results showed that the developed framework is effective to both build solutions and analyze them for a UAS-based delivery system. The results lead to a conclusion that Hypothesis for RQ 4 is demonstrated. These demonstrations showed that Research Objective (RO) has been satisfied, and the developed framework for concurrent unmanned vehicle design and routing for urban delivery systems is suitable to either analyze or design a sUAS-based urban delivery system.

7.2 Future Work

This thesis developed four main modules: vehicle routing optimization model, flight urban network, UAV sizing, and concurrent framework. The first area of future work is extending the vehicle routing model. Specifically, for operations in a larger area, multiple depots can be addressed. The concept of operations of a multi-depot mission is introduced in Appendix A.3. This extension will allow to address a generalized UAS-based delivery system. Other missions rather than a delivery mission, also, can be addressed such as monitoring a target

area as shown in Appendix A.2 and A.3. This is allowed because the developed framework is designed to be flexible in merging the modules.

The second area of future work is merging vehicle dynamics into the developed method to build the two-layered urban flight network to create smooth collision-free paths. The developed method to build an urban flight network creates piece-wise linear collision-free paths. In order to address more precise curves, the developed method can be extended with a path-planning method including vehicle dynamics. For instance, an algorithm to create a Dubins path introduced in Appendix A.3 can be merged.

The next area of future work is integrating a more precise UAV sizing module including vehicle configuration design. This extension will allow to estimate vehicle parameters more precisely. Furthermore, developing a vehicle cost model linked to the sizing module for fixed/operating costs also will be useful.

The final area of future work is extending the developed multi-disciplinary optimization method. The developed method used a fixed-point iteration method. This method can be a baseline for other multi-disciplinary optimization approaches such as Collaborative Optimization (CO) and Analytical Target Cascading (ATC). This extension would either be computationally efficient or address a better solution in a global optimization point of view.

total cost optimized

7.3 Contributions

This thesis provided several contributions to the area of modeling a UAS-based delivery system. The first contribution is presenting both maximum endurance and energy constraints in mathematical formulations for a UAS-based delivery system. In order to address a UAS-based delivery system, one of them should be applied to an optimization model. In this thesis, the constraints are provided as mathematical formulation, which allows to use a commercial solver directly. This enables a rapid development of a mathematical model.

The second contribution is integrating a point cloud based urban model with optimization models. This enables a more precise analysis of a UAS-based delivery system with a realistic problem and allows to apply collision-free paths to an optimization model. This also provides broad expandability of a use of optimization models with accessible/controllable information of a vehicle network in the urban environment.

The final contribution is extending a sizing and synthesis process for sVTOL UAVs by including UAV routing process based on a vehicle routing problem optimization model for sUAS-based delivery systems. The developed framework deals with the UAV design process and UAV routing process as a black box module respectively. Thus, other processes, algorithms, frameworks can be integrated with the developed framework as a module if their interface is matched.

The developed framework for concurrent UAV design and routing for UAS-based urban delivery systems was demonstrated with realistic problems in order to show how it can be utilized for determining the key parameters of a UAS-based delivery system. The developed framework for concurrent UAV design and routing for UAS-based urban delivery systems can address effects of both vehicle design parameters and the concept of operations of UAVs on the delivery systems concurrently.

Appendices

APPENDIX A

THE VEHICLE ROUTING PROBLEMS: APPLICATIONS AND APPROACHES

A.1 UAS-based Package Delivery: an Extended Savings Algorithm¹

This section presents an extended savings algorithm for UAS-based delivery systems that can reflect the UAV's natures described in Section 1.2. The proposed algorithm deals with the multi-trip vehicle routing problem with time windows (MTVRPTW), which is an extended variant of the VRPTW allowing each UAV to have multiple trips or a journey. Based on the graph theory, a MTVRPTW can be described by a graph $\mathcal{G}(\mathcal{N}, \mathcal{A})$ and a set of vehicles \mathcal{V} , where \mathcal{N} is a set of all nodes in \mathcal{G} , and \mathcal{A} is a set of all arcs in \mathcal{G} . A set of nodes, $\mathcal{N} = \{0, 1, \dots, n, n+1\}$, consists of a set of two depots, $depot = \{0, n+1\}$, and a set of n customers, $\mathcal{C} = \{1, 2, \dots, n\}$. Note that the two depots indicate the same depot, but $depot_0$ is used when a vehicle leaves the depot, while $depot_{n+1}$ is utilized when a vehicle arrives at the depot. Each customer \mathcal{C}_i has preferred time to be visited that is called time window $\mathcal{TW}_i = [a_i, b_i]$, which a_i is ready time, and b_i is due time. Also, each customer has service time s_i , which is the total time of landing, delivering, and takeoff. Each arc \mathcal{A}_{ij} has a distance between \mathcal{N}_i and \mathcal{N}_j , \mathcal{D}_{ij} which is symmetric. Finally, each vehicle \mathcal{V}_i can take a journey \mathcal{J}_i which is a combination of trips in \mathcal{T} . With these notions, this section discusses key rules of the proposed method and present the details of the algorithm structure.

A.1.1 Key Rules of the Extended Saving Algorithm

To convert the key characteristics of delivery system using UAVs mentioned in Section 1.2 into a savings algorithm, the rules that can be applied to the algorithm need to be created. First, the properties of small payload capacity, short endurance and recharging of UAVs can be modeled mathematical expressions. Second, the straight path of UAVs can be reflected

¹This section is published in [27].

by the straight flight distance among nodes. Last, by the assumption of BVLOS operations and fully advanced technologies of UAVs, this problem can be handled from the point of view of high-level operations of UAVs-based delivery system. Additionally, in the proposed method, one more characteristic is infused, which comes from a limitation of the VRPTW model not a property of UAS-based delivery system. In the VRPTW models, if a vehicle arrives at a delivery location before the designated time called ready time, then the vehicle should wait until ready time at the place. If this concept applies to UAVs, they could be wait a long time on the ground or even in the air by hovering. Because the long time hovering may occur some safety issue, a rule to prevent a long-time hovering is combined.

The proposed method, thus, is based on five key rules: saving rule, time window rule, maximum hovering rule, maximum endurance rule, and multi-trip rule. Saving rule is the fundamental idea of all savings algorithms, which provides a priority order of a pair of nodes to be combined. Time window rule describes how to select a customer to be visited earlier using preferred time to be visited for each customer. Maximum hovering rule prohibits long waiting at a delivery location. Endurance rules are directly related with UAV's properties, which guarantees that each UAV returns at the depot within UAV's maximum endurance. Lastly, multi-trip rule needs to be to build a journey that is a combination of trips assigned to a UAV.

Saving Rule

The saving implies how much distance can be saved when two trips merge into one trip. For instance, there are two trips that each trip visits a customer respectively as shown in Fig. A.1a. Then, total distance vehicles should move is calculated by

$$\text{Total Distance}_{[\mathcal{T}_1, \mathcal{T}_2]} = \mathcal{D}_{0i} + \mathcal{D}_{i(n+1)} + \mathcal{D}_{0j} + \mathcal{D}_{j(n+1)} = 2\mathcal{D}_{0i} + 2\mathcal{D}_{0j} \quad (\text{A.1})$$

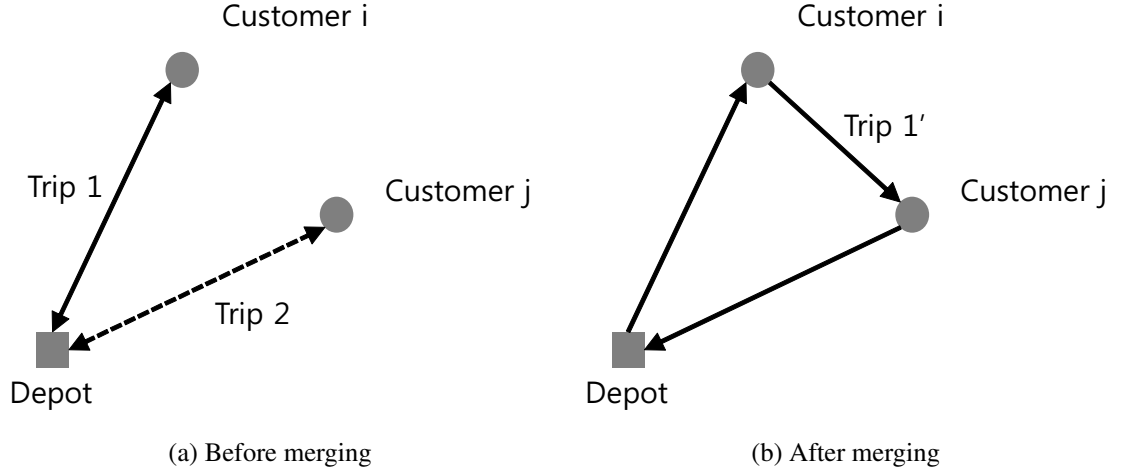


Figure A.1: An example of merging two trips

If the two trips merge to form a new trip like Fig. A.1b, the total distance is updated by

$$\text{Total Distance}_{[\mathcal{T}_{1'}]} = \mathcal{D}_{0i} + \mathcal{D}_{ij} + \mathcal{D}_{j(n+1)} = \mathcal{D}_{0i} + \mathcal{D}_{ij} + \mathcal{D}_{0j} \quad (\text{A.2})$$

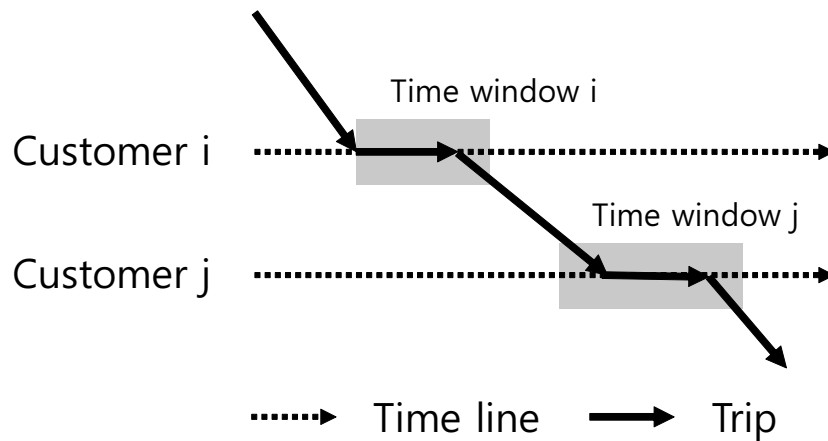
Then, the distance that can reduce by merging the two trips, which is called saving, is computed by

$$\begin{aligned} \mathcal{SV}_{ij} &= \text{Total Distance}_{[\mathcal{T}_1, \mathcal{T}_2]} - \text{Total Distance}_{[\mathcal{T}_{1'}]} \\ &= (2\mathcal{D}_{0i} + 2\mathcal{D}_{0j}) - (\mathcal{D}_{0i} + \mathcal{D}_{ij} + \mathcal{D}_{0j}) \\ &= \mathcal{D}_{0i} + \mathcal{D}_{0j} - \mathcal{D}_{ij}. \end{aligned} \quad (\text{A.3})$$

In the savings algorithm, a pair of trips with high saving value has a priority to merge. Note that when merging trips, at least one of them should be trivial trip which is a trip to visit only one customer.

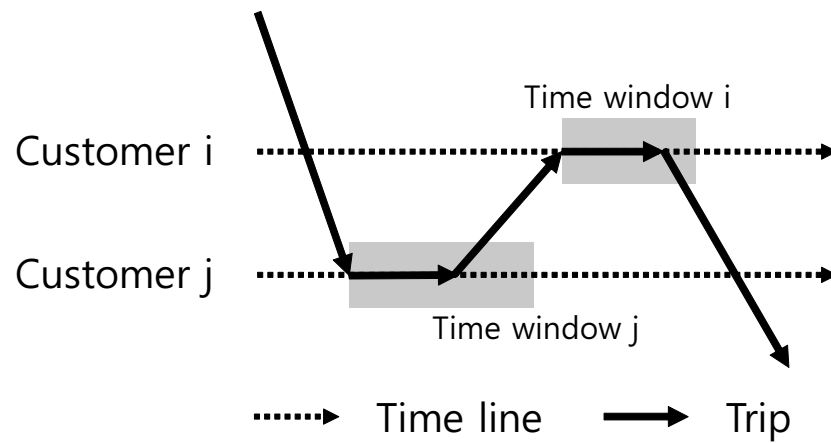
Time Window Rule

In MTRVRPTWs, each customer has preferred time to be visited, which is captured by a time window. The time windows determine the order to visit customers. For example, assume that there are a customer i and customer j who should be visited. Because each customer has own time window, three scenarios could occur: in the first scenario, the time window of the customer i start earlier, and even if a UAV moves from the customer i to the customer j after serving the customer i , it can arrive at the customer j within its time window as shown in Fig. A.2a. In this case, the UAV can serve the two customers on a trip by visiting the customer i earlier. The second scenario is the inverse situation of the first scenario as illustrated in Fig. A.2b. In the same manner, a UAV can serve the two customers with the reverse order to visit the customers. In the last scenario, if the customer j has a time window which it is impossible to be visited after visiting the customer i , and the customer i has a time window which it is also impracticable to be visited after visiting the customer j , then it is unrealizable to allocate two visits for a UAV. Thus, in that case, two trip should remain as shown in Fig. A.2c.

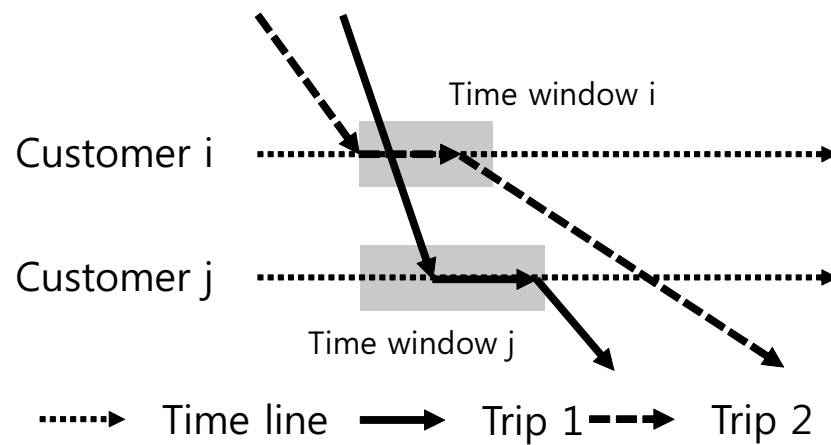


(a) A case that customer i should be visited earlier than customer j

Figure A.2: The effect of time window on the order of visiting customers

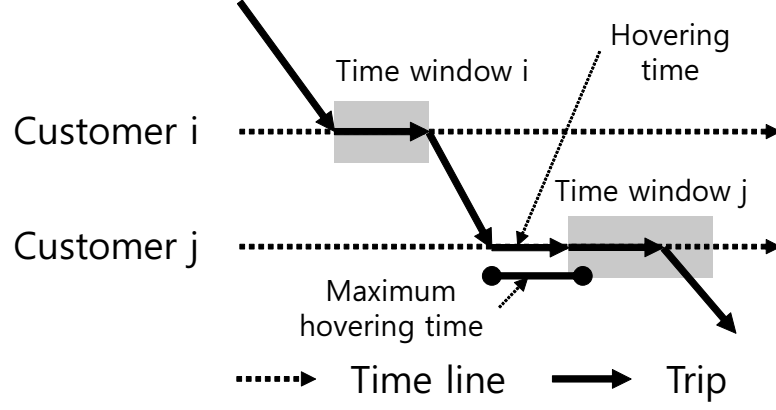


(b) A case that customer j should be visited earlier than customer i

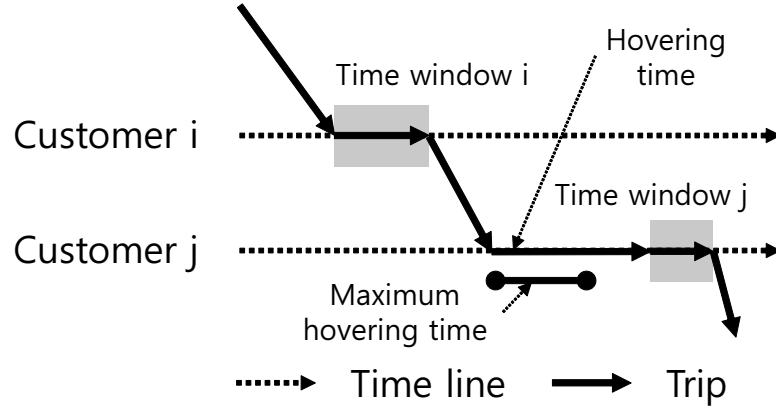


(c) A case that is impossible to merge

Figure A.2: The effect of time window on the order of visiting customers (cont.)



(a) A case of short hovering

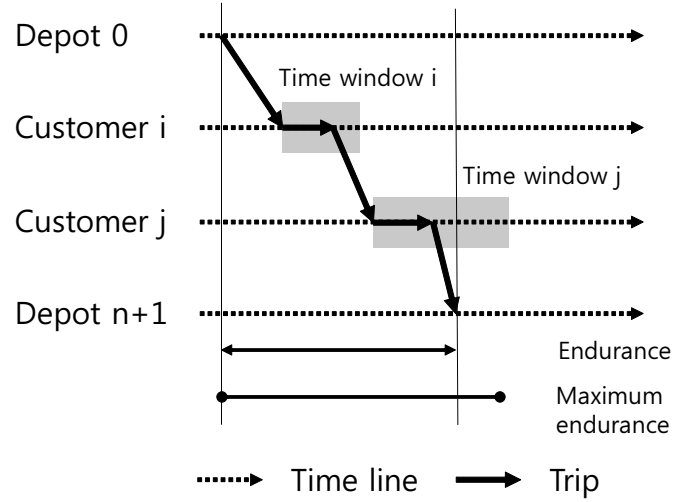


(b) A case of long hovering

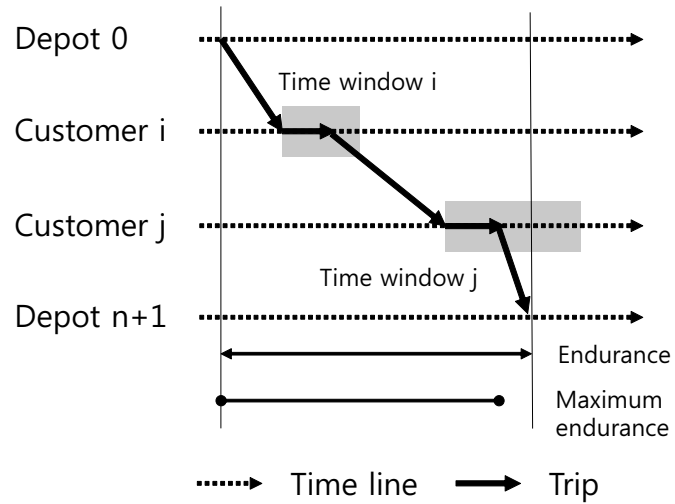
Figure A.3: Maximum hovering rule

Maximum Hovering Rule

Although a UAV can visit two customers on a trip based on their time windows, it is possible that the UAV should stay in the air until ready time of the second customer. In this case, as hovering time increases, a safety issue could be emerged with anxieties of neighbors because of noise and privacy issues. In addition, by restricting UAVs to have long hovering, they are forced to return to the depot more frequently to complete the entire mission, but this could increase the opportunity of merging trips to build a journey. Thus, we assume that if a UAV has longer hovering time to visit a customer than the prefixed maximum hovering time, it cannot visit the customer as illustrated in Fig. A.3.



(a) A feasible case to visit two customers on a trip

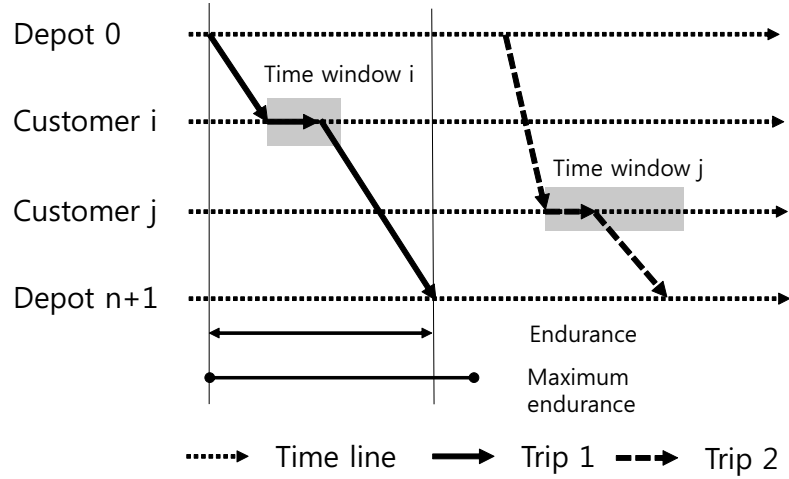


(b) An infeasible case to visit two customers on a trip

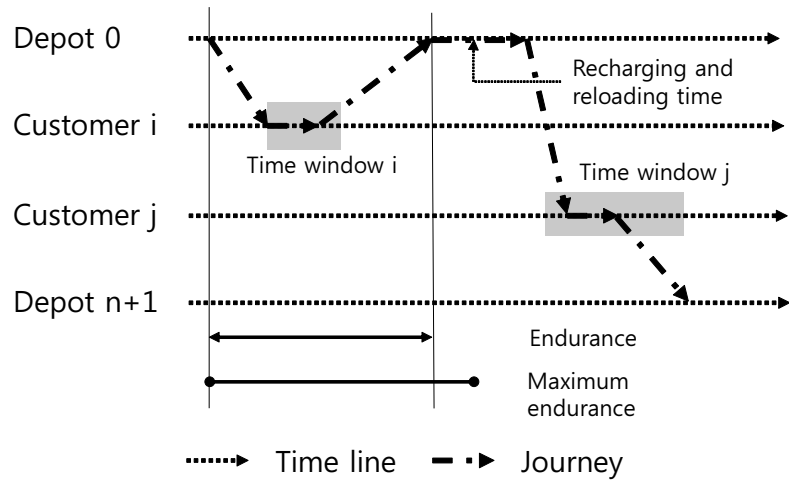
Figure A.4: Maximum endurance rule

Maximum Endurance Rule

In conventional savings algorithms, only payload capacity has used to check whether or not a vehicle can visit a next customer. However, because a UAV has much shorter endurance, it should often visit a location where its battery can be swapped. If the flight time of a trip becomes longer than maximum endurance of the UAV by visiting more customers, the trip should not be extended as an infeasible case as shown in Fig. A.4.



(a) Two trips which can merge to form a journey



(b) A journey which is created by merging two trips

Figure A.5: Multi-trip rule

Multi-Trip Rule

Allowing UAVs to have multiple trips is a key point to significantly reduce the acquisition cost of UAS-based delivery systems. By reusing UAVs, the minimum required number of UAVs decreases, which yields the maximum available number of UAVs concurrently. In order to merge two trips to create a journey, at least there is time enough to reload packages and to swap UAV's batteries as illustrated in Fig. A.5.

A.1.2 The Algorithm Structure of the Extended Savings Algorithm

The extended savings algorithm consists of three steps; computing savings, creating trips, and building journeys. Each step sequentially works from computing savings to building journeys. The first step finds a set of feasible arcs to be capable of merging with their own savings. For feasibility checks, time window rule and maximum hovering rule are applied. If an arc is infeasible, the two customers on the arc cannot be visited sequentially on a trip. For feasible arcs, saving is computed by saving rule, Eq. (A.3). We note that before looking over each arc, if there is the sorted list of customers based on time windows, computational power can be saved.

In the second step, trivial trips need to be made first, which visit just one customer. The set of trivial trips is a solution of the single-package delivery operation. Thus, if the single-package delivery operation is considered, this step can be skipped. After creating trivial trips, the list of arcs based on saving values is sorted, which implies the priority to merge. Then, during looking over each arc sorted by saving values, time window rule, endurance rule, payload capability check are utilized to find a pair of trip that can merge. Note that when merging two trips, at least one of them should be a trivial trip. In the conventional savings algorithms, the algorithm ends by returning trips.

In the last step, journeys which are allocated to each UAV are built from the trips created in the second step. Each journey has at least one trip, because it is a combination of trips. In this step, sorted trip list is utilized as a priority to merge. Finally, during looking over each trip, journeys are built by journey rule. Note that each journey is assigned to each UAV. The procedure of the proposed algorithm is described as pseudo codes in Algorithm 2.

A.1.3 Merge Strategies: Sequential vs. Parallel

When merging two routes into one route, two approaches can be used; sequential and parallel merge strategies. First, the sequential merge strategy works with a single base

Algorithm 2 Pseudo code of the proposed extended savings algorithm

Input: $\mathcal{G}(\mathcal{N}, \mathcal{A})$ and \mathcal{V}

Output: \mathcal{J}

[STEP 1: computing savings using saving, time window, and maximum hovering rules]

Initialization

Sort nodes based on the order of time windows

for i in \mathcal{C} **do**

for j in \mathcal{C} **do**

if the order to be visited and max hovering time constraint are feasible **then**

$$SV_{ij} = D_{0i} + D_{0j} - D_{ij}$$

end if

end for

end for

[STEP 2: creating trips using time window and maximum endurance rules]

Create \mathcal{T} with trivial trips for each customer

Sort \mathcal{A} by descending order based on SV

for (i, j) in \mathcal{A} **do**

 Find trips that visit \mathcal{C}_i and \mathcal{C}_j respectively

if at least either \mathcal{C}_i or \mathcal{C}_j is in a trivial trip **then**

 Estimate time schedule and payload capacity to be used after merging two trips

if time schedule, payload capacity, endurance constraints are feasible **then**

 Merge two trips

end if

end if

end for

[STEP 3: building journeys using time window and multi-trip rules]

\mathcal{J} = empty set

Sort \mathcal{T} based on the order of departure time

for \mathcal{T}_i in \mathcal{T} **do**

for \mathcal{J}_i in \mathcal{J} **do**

if \mathcal{T}_i can be merged into \mathcal{J}_i **then**

 Update \mathcal{J}_i as \mathcal{J}_i with \mathcal{T}_i

end if

end for

if there is no journey which \mathcal{T}_i can be merged into **then**

 Add a journey as \mathcal{T}_i into \mathcal{J}

end if

end for

return \mathcal{J}

route which has the largest saving value. The approach extends the base route as much as possible, then it considers a next base route. Second, the parallel merge strategy unites two routes along the order of a sorted list of saving values without considering a base route. The literature written by Laporte et al. reports that the parallel merge strategy generally creates better routes than the corresponding sequential merge strategy [134].

For instance, let us consider four trivial routes such that $[0, 1, 5]$, $[0, 2, 5]$, $[0, 3, 5]$, and $[0, 4, 5]$, where 0 and 5 indicate a depot, and the others are delivery locations. It is also assumed that a sorted list of saving values is given such that $(1, 2): 10$, $(3, 4): 8$, $(2, 4): 7$, which means when two nodes in parenthesis are merged, the saving value after colon can be obtained. In the sequential merge approach, $[0, 1, 5]$ and $[0, 2, 5]$ routes can be combined into $[0, 1, 2, 5]$ called a base route. Then, the pair of node 3 and 4 in the list is skipped because it cannot be used to extend the base route. Next, the route $[0, 4, 5]$ is merged into the base route. As the result, the algorithm returns $[0, 1, 2, 4, 5]$ and $[0, 3, 5]$ routes as a solution with the total saving value, 17.

On the other hand, in the parallel merge approach, after combining $[0, 1, 5]$ and $[0, 2, 5]$ routes into $[0, 1, 2, 5]$ route, $[0, 3, 4, 5]$ route is created by merging $[0, 3, 5]$ and $[0, 4, 5]$ routes rather than $[0, 1, 2, 4, 5]$ route in the sequential merge approach. This behavior is expected because the saving value from merging $[0, 3, 5]$ and $[0, 4, 5]$ routes is larger than that from combining $[0, 1, 2, 5]$ and $[0, 4, 5]$ routes. That is, the parallel approach tries to maximize total saving value by combining with multiple routes at once along the sorted list of saving values. Next, the algorithm considers to unite two route visiting node 2 and 4 respectively. However, node 2 and 4 cannot be connected because both are not trivial routes. Thus, the parallel approach returns $[0, 1, 2, 5]$ and $[0, 3, 4, 5]$ routes as a solution with total saving value, 18. This example is illustrated in Figure A.6.

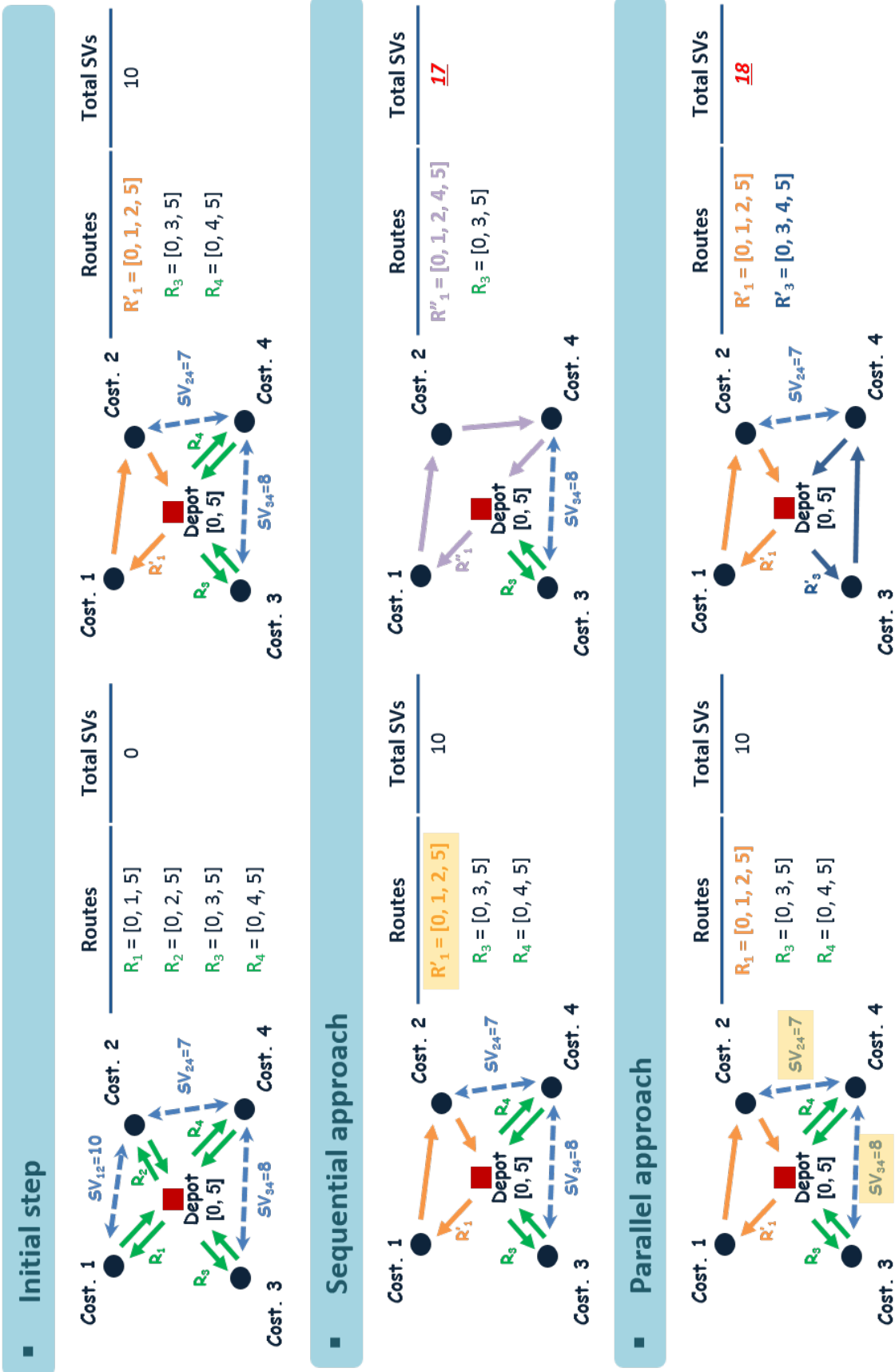


Figure A.6: Comparison of two merge strategies

A.2 Coverage Path Planning: a Column Generation Method ²

A.2.1 Problem Modeling

Coverage path planning problem

The main purpose of solving a multi-UAV-based CPP problem is to build an optimal coverage route/path for each UAV that depends on a shape of an AOI and a size of sensor footprint as illustrated in Fig. A.7. The footprint is determined by an operating altitude and the field of view of a sensor. In the context of the CPP problem, the optimal coverage route is based on back-and-forth routes or spiral routes as shown in Fig. A.8, which requires the minimum number of turns since they can minimize total energy consumption [135, 136].

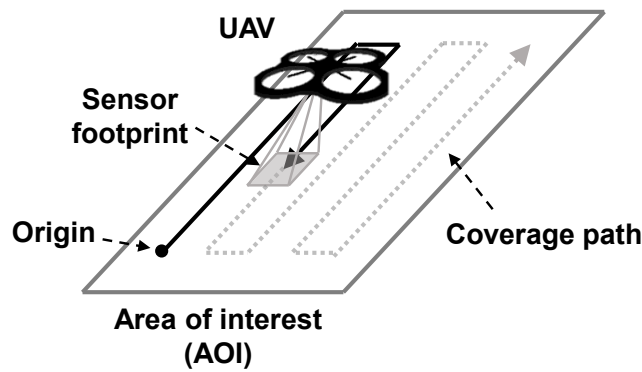


Figure A.7: An illustration of a coverage path planning problem

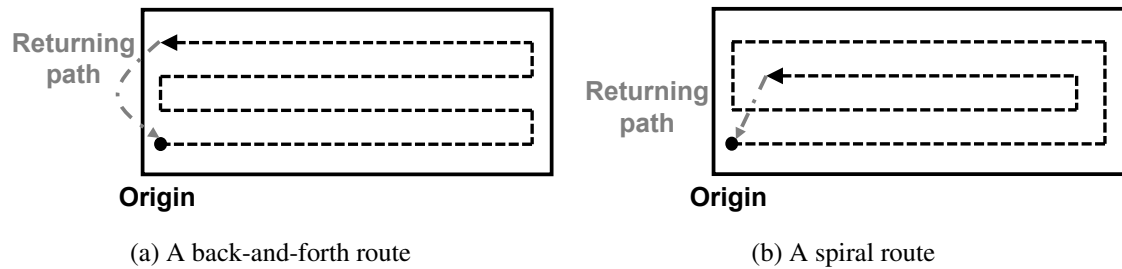


Figure A.8: Preferred routes for the coverage path planning problems

²Previous work of this section is published in [130], and this section is published in [132].

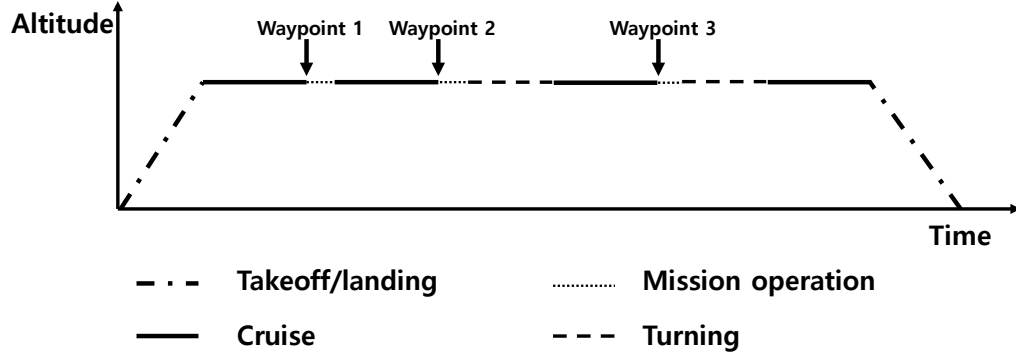


Figure A.9: A mission profile of a UAV for an imagery mission

Estimation of energy consumption

In order to execute a UAS imagery mission, a typical mission profile consists of a takeoff, cruise, and hovering to obtain imagery data at each waypoint, turning, and landing. For instance, a mission profile with 3 waypoints is described in Fig. A.9. The CPP flying trajectory may include several turning phases right after acquiring imagery data to reach the next waypoint or terminal depot position. For the simplified energy estimation, those five mission segments are categorized into two flight conditions: hovering and forward flight. The hovering phase entails takeoff/landing/turning/image acquisition segments, and the forward flight phase has cruise segment.

For multicopters, hovering power is larger than takeoff/landing power at a low climb/descent speed in the normal working state or the windmill brake state [120]. A hovering, also, is required to acquire imagery data as well as to turn its body to head for a next waypoint or a depot. Thus, we use hovering power for mission segments except for cruise one. Based on the momentum theory [120], the power required to hover is described as

$$P_h = T \sqrt{\frac{T}{2\rho A}} = \frac{T^{3/2}}{\sqrt{2\rho A}} = \frac{W^{3/2}}{\sqrt{2\rho A}} \quad (\text{A.4})$$

where T is the thrust required to hover, W is the weight of UAV, ρ is the density of the air, and A is the disk area. For multicopters having n rotors, the ideal power required to hover

for each rotor can be calculated by

$$P_{h,(1/n)} = \frac{(W/n)^{3/2}}{\sqrt{2\rho(A/n)}} \quad (\text{A.5})$$

To estimate more practical power, the Figure of Merit (FM) is commonly used, if the information is available:

$$FM = \frac{P_{ideal}}{P_{actual}} < 1 \quad (\text{A.6})$$

In a cruise mission segment, the thrust required in forward flight for each rotor can be calculated as

$$T_{(1/n)} = 2\rho A_{(1/n)} v_i \sqrt{(V_\infty \cos \alpha)^2 + (V_\infty \sin \alpha + v_i)^2} \quad (\text{A.7})$$

where V_∞ is a free stream velocity, α is an angle of attack, and v_i is an induced velocity. Then, from Eq. (A.7), the induced velocity in forward flight becomes

$$v_i = \frac{v_h^2}{\sqrt{(V_\infty \cos \alpha)^2 + (V_\infty \sin \alpha + v_i)^2}} \quad (\text{A.8})$$

where $v_h^2 = T_{(1/n)}/2\rho A_{(1/n)}$ is the induced velocity in the hovering phase. Using the advance ratio $\mu = V_\infty \cos \alpha / \Omega R$, where Ω is the rotational speed of a rotor, and R is the rotor radius, the inflow ratio is given by

$$\lambda = \frac{V_\infty \sin \alpha + v_i}{\Omega R} = \frac{V_\infty \sin \alpha}{\Omega R} + \frac{v_i}{\Omega R} = \mu \tan \alpha + \lambda_i \quad (\text{A.9})$$

where, λ_i is the induced velocity ratio in forward flight. Then, Eq. (A.8) can be written as

$$\lambda_i = \frac{\lambda_h^2}{\sqrt{\mu^2 + \lambda^2}} = \frac{C_T}{2\sqrt{\mu^2 + \lambda^2}} \quad (\text{A.10})$$

where $\lambda_h = \sqrt{C_T/2}$ from the hovering phase, and C_T is the rotor thrust coefficient, $C_T =$

$T/\rho A(\Omega R)^2$. By substituting Eq. (A.10) for λ_i in Eq. (A.9), the inflow ratio is given by

$$\lambda = \mu \tan \alpha + \frac{C_T}{2\sqrt{\mu^2 + \lambda^2}} \quad (\text{A.11})$$

Once a solution of Eq. (A.11) is obtained, the power required in forward flight is calculated as

$$P_f = P_h \frac{\lambda}{\lambda_h} \quad (\text{A.12})$$

In order to solve Eq. (A.11) to obtain λ , a Newton-Raphson iteration can be used [120]. To use the iterative method, Eq. (A.11) can be written as an iteration equation with iteration number n such as

$$\lambda_{n+1} = \mu \tan \alpha + \frac{C_T}{2\sqrt{\mu^2 + \lambda_n^2}} \quad (\text{A.13})$$

Then, the iteration scheme is given by

$$\lambda_{n+1} = \lambda_n - f(\lambda_n)/f'(\lambda_n) \quad (\text{A.14})$$

where $f(\lambda_n)$ and $f'(\lambda_n)$ are defined as

$$f(\lambda_n) = \lambda_n - \mu \tan \alpha - \frac{C_T}{2\sqrt{\mu^2 + \lambda_n^2}} = 0 \quad (\text{A.15})$$

and

$$f'(\lambda_n) = 1 + \frac{C_T \lambda_n}{2} (\mu^2 + \lambda_n^2)^{-3/2} \quad (\text{A.16})$$

respectively. This iteration stops when

$$\epsilon = \left\| \frac{\lambda_{n+1} - \lambda_n}{\lambda_{n+1}} \right\| < 0.0005 = 0.05\% \quad (\text{A.17})$$

A.2.2 Optimization Model

A VRP is a kind of Mixed Integer Linear Programming (MILP) problems including both real variables and integer variables with a linear objective function and linear constraints. The VRP determines optimal routes for each vehicle which are modeled by integer variables. Each route is constrained by vehicle characteristics such as its payload capacity and maximum range. The VRP models utilize arcs or routes as design variables to model routes, which are called an arc-based optimization model and a route-based optimization model, respectively. To address CPP problems, we propose an arc-based and energy-constrained optimization model (ABECOM) which is an extension of arc-based and distance-constraint model [130, 137], and then the model is converted to a route-based and energy-constrained optimization model (RBECOM) to impose an effect of turning motions on the energy-based cost function.

Arc-based optimization model for energy-constrained CPP problems

An ABECOM is described by an input graph describing a vehicle network, $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, and a fleet of vehicles, $\mathcal{V} = \{1, \dots, l\}$. A graph, \mathcal{G} , contains a set of nodes, \mathcal{N} , and a set of arcs, \mathcal{A} : a set of nodes, $\mathcal{N} = \{0, 1, 2, \dots, n, n+1\}$, consists of the starting depot, 0, the returning depot, $n+1$, and waypoints, $\mathcal{W} = \{1, 2, \dots, n\}$. Let \mathcal{N}_O be the starting depot and waypoints, $\mathcal{N}_O = \{0, 1, 2, \dots, n\}$, and \mathcal{N}_D be waypoints and the returning depot, $\mathcal{N}_D = \{1, 2, \dots, n, n+1\}$, then a set of arcs is defined as $\mathcal{A} = \{(i, j) : \forall i \in \mathcal{N}_O, \forall j \in \mathcal{N}_D, i \neq j\}$. In a UAS imagery mission, the locations of waypoints, \mathcal{W} , are selected by image sensor's footprint that depends on both a sensor specifications and a mission altitude. A cost coefficient c_{ij}^k of each arc corresponds with energy required for all operations such as a forward flight from node i to node j by vehicle k , $(i, j) \in \mathcal{A}$ and $k \in \mathcal{V}$, mission operations at node j , $j \in \mathcal{W}$, a takeoff, $i = 0$ and $j \in \mathcal{W}$, and a landing, $i \in \mathcal{W}$ and $j = n+1$. Each vehicle has maximum energy, \mathcal{E}^k , available to use during a flight, $k \in \mathcal{V}$. For an arc-based model, a design variable, x_{ij}^k , is defined such that if vehicle k , $k \in \mathcal{V}$,

travels along an arc (i, j) , $(i, j) \in \mathcal{A}$, then x_{ij}^k is defined as 1, otherwise 0. With these definitions, an ABECOM can be written by

$$\begin{aligned} &\text{Minimize} && \sum_{k \in \mathcal{V}} \sum_{(i,j) \in \mathcal{A}} c_{ij}^k x_{ij}^k \end{aligned} \quad (\text{A.18})$$

Subject to

$$\sum_{k \in \mathcal{V}} \sum_{j \in \mathcal{N}_D} x_{ij}^k = 1 \quad (\forall i \in \mathcal{W}) \quad (\text{A.19})$$

$$\sum_{j \in \mathcal{N}_D} x_{0j}^k = 1 \quad (\forall k \in \mathcal{V}) \quad (\text{A.20})$$

$$\sum_{i \in \mathcal{N}_O} x_{ih}^k - \sum_{j \in \mathcal{N}_D} x_{hj}^k = 0 \quad (\forall h \in \mathcal{W}, \forall k \in \mathcal{V}) \quad (\text{A.21})$$

$$\sum_{i \in \mathcal{N}_O} x_{i(n+1)}^k = 1 \quad (\forall k \in \mathcal{V}) \quad (\text{A.22})$$

$$\begin{aligned} \sum_{j \in \mathcal{N}_D} s_{hj}^k - \sum_{i \in \mathcal{N}_O} s_{ih}^k - \sum_{j \in \mathcal{N}_D} (e_F^k d_{hj}^k + e_M^k t_j^k) x_{hj}^k &= 0 \\ &(\forall h \in \mathcal{W}, \forall k \in \mathcal{V}) \end{aligned} \quad (\text{A.23})$$

$$\begin{aligned} s_{0j}^k &= (e_F^k d_{0j}^k + e_M^k t_j^k + e_T^k t_T^k) x_{0j}^k \quad (\forall j \in \mathcal{N}_D, \\ &\forall k \in \mathcal{V}) \end{aligned} \quad (\text{A.24})$$

$$\begin{aligned} s_{ih}^k &\leq (\mathcal{E}^k - e_F^k d_{h(n+1)}^k - e_L^k t_L^k) x_{ih}^k \quad (\forall i \in \mathcal{N}_O, \forall h \in \mathcal{W}, \\ &\forall k \in \mathcal{V}) \end{aligned} \quad (\text{A.25})$$

$$s_{i(n+1)}^k \leq \mathcal{E}^k x_{i(n+1)}^k \quad (\forall i \in \mathcal{N}_O, \forall k \in \mathcal{V}) \quad (\text{A.26})$$

$$\begin{aligned} s_{hj}^k &\geq (e_F^k (d_{0h}^k + d_{hj}^k) + e_M^k (t_h^k + t_j^k) + e_T^k t_T^k) x_{hj}^k \\ &(\forall h \in \mathcal{W}, \forall j \in \mathcal{N}_D, \forall k \in \mathcal{V}) \end{aligned} \quad (\text{A.27})$$

$$x_{ij}^k \in \{0, 1\} \quad (\forall (i, j) \in \mathcal{A}, \forall k \in \mathcal{V}) \quad (\text{A.28})$$

where s_{ij}^k is a flow variable that traces energy consumed by vehicle k from the starting depot to node j , $(i, j) \in \mathcal{A}$, $k \in \mathcal{V}$, and d_{ij}^k is a flight distance from node i to node j by vehicle k , $(i, j) \in \mathcal{A}$ and $k \in \mathcal{V}$, and e_F^k is energy per unit distance required by vehicle k , $k \in \mathcal{V}$, for a forward flight, and e_M^k is energy per unit time required by vehicle k , $k \in \mathcal{V}$, for mission operations, and t_j^k/t_h^k is time required by vehicle k for mission operations at node j/h , $j \in \mathcal{N}_D$, $h \in \mathcal{W}$, $k \in \mathcal{V}$. The variable e_T^k/e_L^k is energy per unit time required by vehicle k for a takeoff/landing, and t_T^k/t_L^k is time required by vehicle k for a takeoff/landing respectively.

The objective function of the ABECOM, Eq. (A.18), is the total energy required for a given mission which is proportional to a total flight distance of the UAVs. Operational strategies for an aerial imaging mission are treated as constraints. Every waypoint should be visited exactly once by a UAV, Eq. (A.19). Each route should start and end at the depot, Eq. (A.20) and (A.22). If a UAV visits a waypoint, then it should leave for another place, Eq. (A.21). The Sub-tour elimination condition is modeled as Eq. (A.23). The initial required energy for each UAV is defined by Eq. (A.24). The upper bounds of energy consumed at intermediate and final parts of each route are constrained by Eq. (A.25) and (A.26) respectively whereas the lower bound of energy consumed up to node j is restricted by Eq. (A.27). The integrity constraint of x_{ij}^k is shown in Eq. (A.28).

Route-based optimization model for energy-constrained CPP problems

The ABECOM searches an optimal solution with minimum energy consumption to cover an entire AOI. However, imposing turning impacts based on the ABECOM is very challenging because of the limitation of arc-based formulation [130]. Before directly dealing with the limitation, we present the RBECOM comprised of a master problem and subproblems in this subsection, which is an equivalent reformulation of the ABECOM.

A standard MILP problem including n design variables and m constraints can be written

as a matrix expression such that

$$\text{Minimize} \quad \mathbf{c}^T \mathbf{x} \quad (\text{A.29})$$

$$\text{Subject to} \quad \mathbf{A}\mathbf{x} \geq \mathbf{b} \quad (\text{A.30})$$

$$\mathbf{x} \geq 0 \quad (\text{A.31})$$

$$\mathbf{x} \in \mathbb{Z}^p \times \mathbb{R}^{n-p} \quad (\text{A.32})$$

where \mathbf{c}^T is a cost coefficient vector, $\mathbf{c}^T \in \mathbb{R}^n$, and \mathbf{A}/\mathbf{b} are a constraint matrix/vector, $\mathbf{A} \in \mathbb{R}^{n \times m}$ and $\mathbf{b} \in \mathbb{R}^m$, and \mathbf{x} is a design variable vector consisting of p integer numbers and $n-p$ real numbers. A constraint matrix of a VRP has a block angular structure such that some constraints are applied to whole design variables whereas the others have an effect on a subset of the design variables. The block angular structure system can be decomposed by the Dantzig-Wolfe decomposition (DWD) [138] into two problems: coupled and non-coupled problems. Note that in the ABECOM, only the constraint that every customer is visited by a UAV, Eq. (A.19), is coupled with vehicles, \mathcal{V} . Thus, the ABECOM can be decomposed into a vehicle-coupled problem called a master problem and l vehicle-non-coupled problems called subproblems as shown in Fig. A.10. Given $|\cdot|$ is the number of elements of a set, $l = |\mathcal{V}|$.

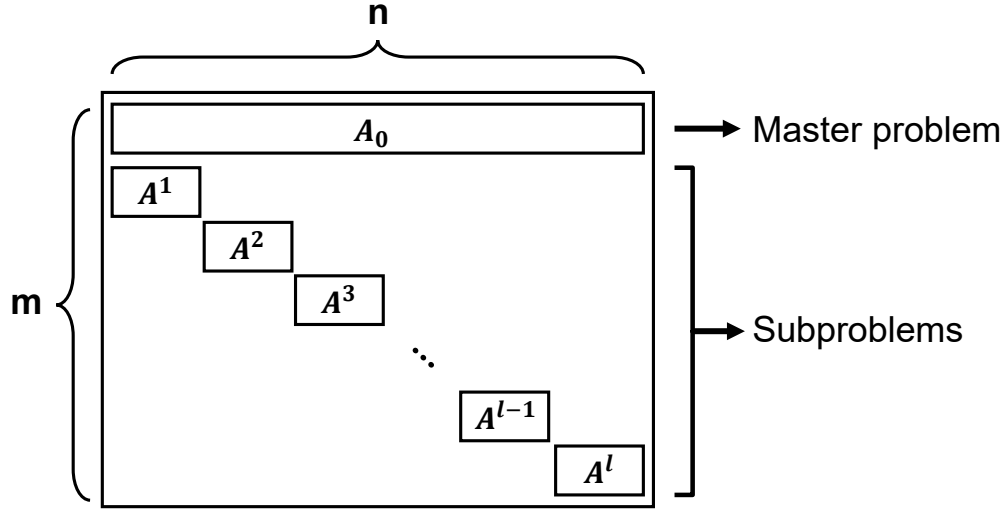


Figure A.10: Block angular structure of a constraint matrix

The master problem The master problem of the ABECOM consists of Eq. (A.18), (A.19), and (A.28), which is a convex combinatorial optimization problem. The master problem can be reformulated by the Minkowski-Weyl theorem; a matrix \mathbf{A} and a vector \mathbf{b} such that $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}\}$ exist. That is, \mathcal{S} is a polyhedron, if and only if there is $\mathbf{B} \in \mathbb{R}^{n \times p}$ and $\mathbf{C} \in \mathbb{R}^{n \times q}$ such that $\mathcal{S} = \text{conv}(\mathbf{B}) + \text{cone}(\mathbf{C})$ [139]. In network flow problems, an extreme point of the polytope in its design space corresponds to a route in the network [140]. Thus, the arc-based design variables of the master problem in the ABECOM can be stated as the convex combination of route-based design variables. Let \mathcal{P}^k be a set of feasible routes for vehicle k , $k \in \mathcal{V}$, and x_{ijp}^k be 1 if vehicle k , $k \in \mathcal{V}$, travels along arc (i, j) , $(i, j) \in \mathcal{A}$, on route p , $p \in \mathcal{P}^k$, otherwise 0. Let y_p^k be the number of times vehicle k , $k \in \mathcal{V}$, travels on route p , $p \in \mathcal{P}^k$. The arc-based design variable x_{ij}^k can be described as

$$\begin{aligned}
 x_{ij}^k &= \sum_{p \in \mathcal{P}^k} x_{ijp}^k y_p^k \quad (\forall (i, j) \in \mathcal{A}, \forall k \in \mathcal{V}), \\
 \sum_{p \in \mathcal{P}^k} y_p^k &= 1 \quad (\forall k \in \mathcal{V}), \\
 y_p^k &\geq 0 \quad (\forall p \in \mathcal{P}^k, \forall k \in \mathcal{V})
 \end{aligned} \tag{A.33}$$

Using route-based design variables, an integer master problem (IMP) can be expressed as

$$f_{IMP} := \text{Minimize } \sum_{k \in \mathcal{V}} \sum_{p \in \mathcal{P}^k} \sum_{(i,j) \in \mathcal{A}} c_{ij}^k x_{ijp}^k y_p^k \quad (\text{A.34})$$

Subject to

$$\sum_{k \in \mathcal{V}} \sum_{p \in \mathcal{P}^k} \sum_{j \in \mathcal{N}_D} x_{ijp}^k y_p^k = 1 \quad (\forall i \in \mathcal{W}) \quad (\text{A.35})$$

$$\sum_{p \in \mathcal{P}^k} y_p^k = 1 \quad (\forall k \in \mathcal{V}) \quad (\text{A.36})$$

$$y_p^k \geq 0 \quad (\forall p \in \mathcal{P}^k, \forall k \in \mathcal{V}) \quad (\text{A.37})$$

$$\sum_{p \in \mathcal{P}^k} x_{ijp}^k y_p^k = x_{ij}^k \quad (\forall (i,j) \in \mathcal{A}, \forall k \in \mathcal{V}) \quad (\text{A.38})$$

$$x_{ij}^k \in \{0, 1\} \quad (\forall (i,j) \in \mathcal{A}, \forall k \in \mathcal{V}) \quad (\text{A.39})$$

From the relation between the arc-based design variables and the route-based design variables, Eq. (A.33), two convexity constraints are added: an affine constraint, Eq. (A.36) and a conic constraint, Eq. (A.37). Linking constraints between the arc-based and route-based variables are stated in Eq. (A.38). To noticeably address the properties of routes, a cost coefficient for each route, c_p^k , and the number of times vehicle k visits customer i , a_{ip}^k , can be defined as

$$c_p^k := \sum_{(i,j) \in \mathcal{A}} c_{ij}^k x_{ijp}^k \quad (\forall p \in \mathcal{P}^k, \forall k \in \mathcal{V}), \quad (\text{A.40})$$

$$a_{ip}^k := \sum_{j \in \mathcal{N}_D} x_{ijp}^k \quad (\forall i \in \mathcal{N}_O, \forall p \in \mathcal{P}^k, \forall k \in \mathcal{V}) \quad (\text{A.41})$$

With route-based terms, the IMP of a RBECOM can be rewritten as

$$f_{IMP} := \text{Minimize } \sum_{k \in \mathcal{V}} \sum_{p \in \mathcal{P}^k} c_p^k y_p^k \quad (\text{A.42})$$

Subject to

$$\sum_{k \in \mathcal{V}} \sum_{p \in \mathcal{P}^k} a_{ip}^k y_p^k = 1 \quad (\forall i \in \mathcal{W}) \quad (\text{A.43})$$

$$\sum_{p \in \mathcal{P}^k} y_p^k = 1 \quad (\forall k \in \mathcal{V}) \quad (\text{A.44})$$

$$y_p^k \geq 0 \quad (\forall p \in \mathcal{P}^k, \forall k \in \mathcal{V}) \quad (\text{A.45})$$

$$\sum_{p \in \mathcal{P}^k} x_{ijp}^k y_p^k = x_{ij}^k \quad (\forall (i, j) \in \mathcal{A}, \forall k \in \mathcal{V}) \quad (\text{A.46})$$

$$x_{ij}^k \in \{0, 1\} \quad (\forall (i, j) \in \mathcal{A}, \forall k \in \mathcal{V}) \quad (\text{A.47})$$

The IP problem can be solved like LP problems with a branch-and-bound approach that is a framework to systematically solve IP problems using a tree data structure. In a branch-and-bound approach, an IP optimization model is commonly transformed to an LP model using a linear relaxation. We note that the LP solution is not guaranteed to be feasible for the IP model. To address this issue, a branch-and-bound approach provides a means for an optimality check of the solution for its IP model. A LP-relaxed master problem (LMP) is stated as

$$f_{LMP} := \text{Minimize } \sum_{k \in \mathcal{V}} \sum_{p \in \mathcal{P}^k} c_p^k y_p^k \quad (\text{A.48})$$

Subject to

$$\sum_{k \in \mathcal{V}} \sum_{p \in \mathcal{P}^k} a_{ip}^k y_p^k = 1 \quad (\forall i \in \mathcal{W}) \quad (\text{A.49})$$

$$\sum_{p \in \mathcal{P}^k} y_p^k = 1 \quad (\forall k \in \mathcal{V}) \quad (\text{A.50})$$

$$y_p^k \geq 0 \quad (\forall p \in \mathcal{P}^k, \forall k \in \mathcal{V}). \quad (\text{A.51})$$

By relaxing Eq. (A.47), there is no longer a need of linking constraints, Eq. (A.46)

The subproblem After implementing the DWD, the RBECOM has l identical subproblems which are independent on vehicles, $l = |\mathcal{V}|$. Each subproblem solves a shortest path problem with resource constraints (SPPRC), which is written as

$$c^k := \text{Minimize} \quad \sum_{(i,j) \in \mathcal{A}} (c_{ij}^k - \pi_i) x_{ij}^k - \pi_0^k \quad (\text{A.52})$$

Subject to

$$\sum_{j \in \mathcal{N}_D} x_{0j}^k = 1 \quad (\text{A.53})$$

$$\sum_{i \in \mathcal{N}_O} x_{ih}^k - \sum_{j \in \mathcal{N}_D} x_{hj}^k = 0 \quad (\forall h \in \mathcal{W}) \quad (\text{A.54})$$

$$\sum_{i \in \mathcal{N}_O} x_{i(n+1)}^k = 1 \quad (\text{A.55})$$

$$\sum_{j \in \mathcal{N}_D} s_{hj}^k - \sum_{i \in \mathcal{N}_O} s_{ih}^k - \sum_{j \in \mathcal{N}_D} (e_F^k d_{hj}^k + e_M^k t_j^k) x_{hj}^k = 0 \quad (\forall h \in \mathcal{W}) \quad (\text{A.56})$$

$$s_{0j}^k = (e_F^k d_{0j}^k + e_M^k t_j^k + e_T^k t_T^k) x_{0j}^k \quad (\forall j \in \mathcal{N}_D) \quad (\text{A.57})$$

$$s_{ih}^k \leq (\mathcal{E}^k - e_F^k d_{h(n+1)}^k - e_L^k t_L^k) x_{ih}^k \quad (\forall i \in \mathcal{N}_O, \forall h \in \mathcal{W}) \quad (\text{A.58})$$

$$s_{i(n+1)}^k \leq \mathcal{E}^k x_{i(n+1)}^k \quad (\forall i \in \mathcal{N}_O) \quad (\text{A.59})$$

$$s_{hj}^k \geq (e_F^k(d_{0h}^k + d_{hj}^k) + e_M^k(t_h^k + t_j^k) + e_T^k t_T^k) x_{hj}^k \quad (\forall h \in \mathcal{W}, \forall j \in \mathcal{N}_D) \quad (\text{A.60})$$

$$x_{ij}^k \in \{0, 1\} \quad (\forall (i, j) \in \mathcal{A}) \quad (\text{A.61})$$

where π_i and π_0^k are dual variables from Eq. (A.49) and (A.50) respectively. Constraints Eq. (A.53) - (A.55) describe vehicle flow from/to the depot. Subtour elimination constraint can be expressed by Eq. (A.56). Energy consumed by each vehicle is constrained by Eq. (A.57) - (A.60). The integrity constraints of x_{ij}^k are stated in Eq. (A.61).

A.2.3 Column Generation of the RBECOM for CPP Problems

The VRP is typically modeled by a MILP optimization model. Column generation is an approach to systematically obtain a solution of an LP problem. Its fundamental ideas are presented with the DWD [138] which decomposes an optimization model with a block-angular-structural constraint matrix into a master problem and subproblems as described in Section A.2.2. When solving practical problems using column generation, a physical meaning of each extreme point needs to be clear. For instance, the first practical problem solved by column generation is a cutting stock problem [141, 142]. Each extreme point of the cutting stock problem indicates a cutting pattern that is how to cut a stock into pieces of specified size. In network flow problems, each extreme point means a flow pattern, a path/route. With this physical meaning, the column generation has been utilized to solve network flow problems such as urban transit crew scheduling problems [143] and vehicle routing problems with time windows [144]. Recently, the column generation becomes a main approach to handle huge IP problems in logistics and operational research areas

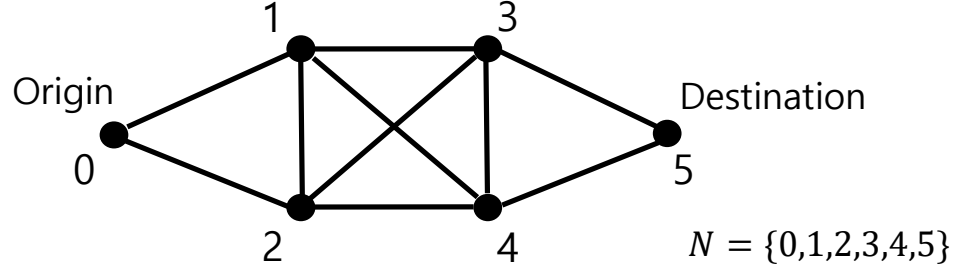


Figure A.11: A simple network topology consisting of 6 nodes and 10 edges

[145]. For UAS-based surveillance missions, the VRPs have been solved by using the column generation method [146, 147].

The extreme point of the CPP problem represents a route that is used as a design variable, y_p^k , in the master problem. The number of y_p^k depends on both the number of vehicles, $|\mathcal{V}|$, and routes, $|\mathcal{P}|$. The RBECOM could need more computing power than the ABECOM even if solving the same problem. For instance, let's assume a simple network topology illustrated in Fig. A.11. The objective of this problem is to find an optimal route from the origin to the destination. To handle this problem, the ABECOM calls for 10 design variables while an RBECOM requires 20 ones. This is because the network topology consists of 10 arcs whereas the number of feasible combinations in the given topology is 20.

To address this computational drawback, one approach is to solve a route-based optimization problem including a subset of routes instead of the whole possible routes such that $\mathcal{P}' \subset \mathcal{P}$ which is called a restricted problem. Let P'_0 be the initial subset of routes, and P'_i be a subset of routes for i -th iteration as illustrated in Fig. A.12. The column generation expands the design space of design variables on every iteration by solving a restricted LP-relaxed master problem and its l subproblems. It stops based on the existence of profitable routes. The column generation framework is shown in Fig. A.13. Each step is described below.

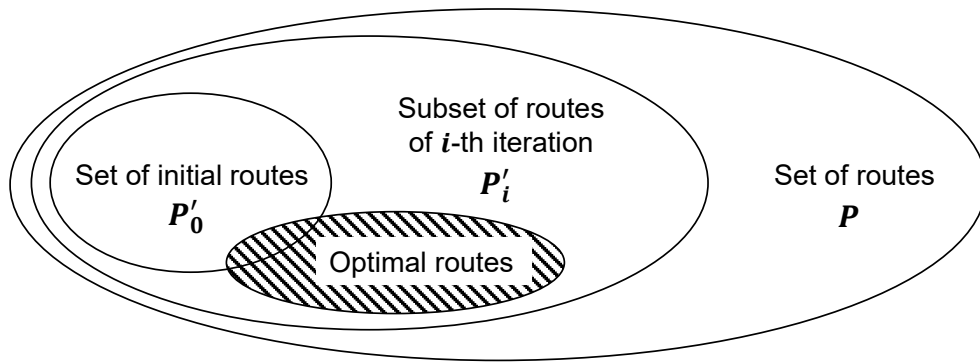


Figure A.12: Change in the domain of routes during the column generation process

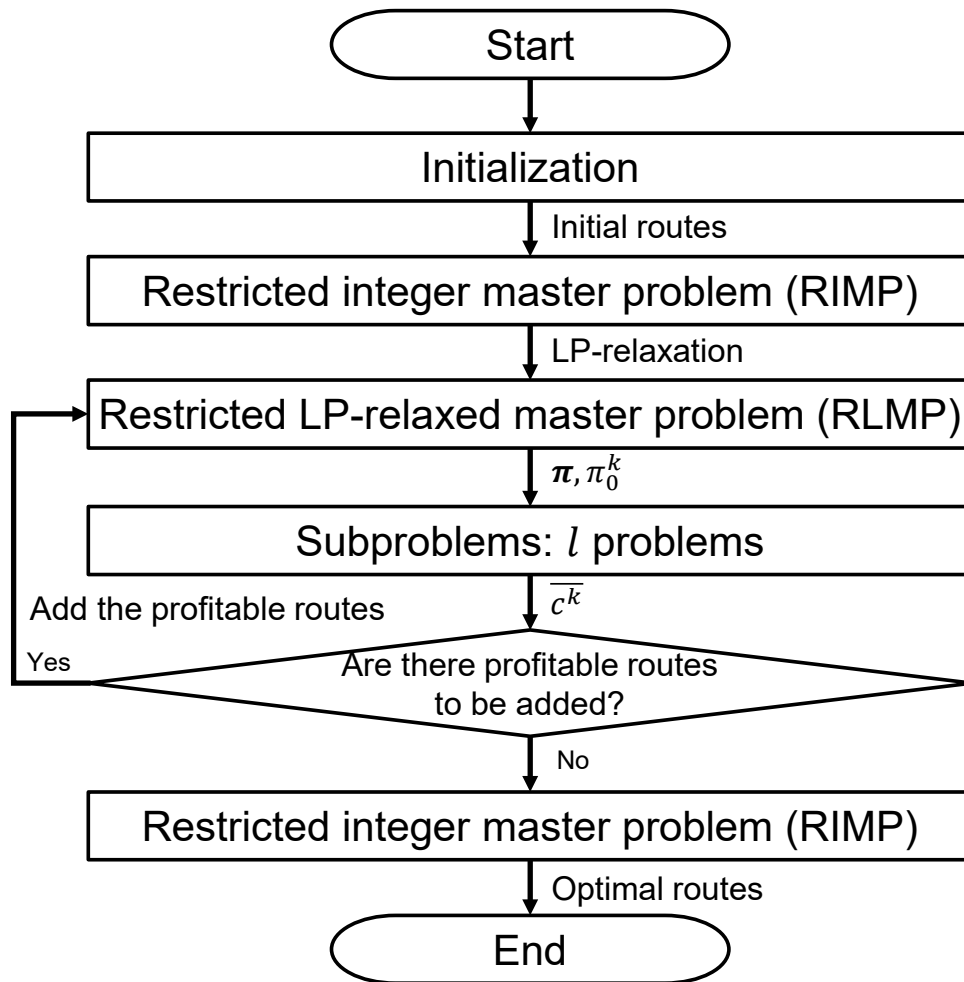


Figure A.13: Column generation framework

Initialization

The initial step of the column generation creates initial solution routes for a CPP problem which are a subset of feasible extreme points. Finding initial feasible routes is commonly solved by two methods: the trivial solution method and the savings algorithm [60]. The trivial solution method creates routes such that each route visits just one waypoint (depot-waypoint-depot routes). The savings algorithm is a greedy method as a simple heuristic approach. Note that the number of initial feasible routes can be used to fix the number of vehicles, $|\mathcal{V}|$. In the RBECOM, the number of vehicles determines the number of subproblems.

Master Problem

The master problem can be modeled by either a set partitioning problems or a set covering problem that is a relaxation of a set partitioning problem [145, 138]. The master problem as a set partitioning problem only considers elementary routes such that each waypoint is visited exactly once. In contrast, the master problem as a set covering problem for CPP problems allows non-elementary routes where each waypoint is visited at least once. In order to find an exact solution of a CPP problem, this authors employ the set partitioning master problem.

The column generation works with a restricted master problem having a subset of routes, $\mathcal{P}' \subset \mathcal{P}$. The solution routes from the initialization step are injected to a restricted integer master problem (RIMP) as an initial subset of routes, $\mathcal{P}'_0 \subset \mathcal{P}$. The RIMP is defined as

$$\begin{aligned} f_{RIMP} := & \text{Minimize} \quad \sum_{k \in \mathcal{V}} \sum_{p \in \mathcal{P}'^k} c_p^k y_p^k & (\text{A.62}) \\ & \text{Subject to} \end{aligned}$$

$$\sum_{k \in \mathcal{V}} \sum_{p \in \mathcal{P}'^k} a_{ip}^k y_p^k = 1 \quad (\forall i \in \mathcal{W}) \quad (\text{A.63})$$

$$\sum_{p \in \mathcal{P}'^k} y_p^k = 1 \quad (\forall k \in \mathcal{V}) \quad (\text{A.64})$$

$$y_p^k \geq 0 \quad (\forall p \in \mathcal{P}'^k, \forall k \in \mathcal{V}) \quad (\text{A.65})$$

$$\sum_{p \in \mathcal{P}'^k} x_{ijp}^k y_p^k = x_{ij}^k \quad (\forall (i, j) \in \mathcal{A}, \forall k \in \mathcal{V}) \quad (\text{A.66})$$

$$x_{ij}^k \in \{0, 1\} \quad (\forall (i, j) \in \mathcal{A}, \forall k \in \mathcal{V}) \quad (\text{A.67})$$

In order to solve the IMP, Eq. (A.42) - (A.47), by using the RIMP, Eq. (A.62) - (A.67), there is a need to find profitable routes for decreasing the objective function value. The process to find the routes is similar with iterations of the simplex method using a non-basic variable to price out and enter the basis. In order for a pricing process of the column generation, the restricted LP-relaxed mater problem (RLMP) is exploited, which is defined as

$$f_{RLMP} := \text{Minimize} \quad \sum_{k \in \mathcal{V}} \sum_{p \in \mathcal{P}'^k} c_p^k y_p^k \quad (\text{A.68})$$

Subject to

$$\sum_{k \in \mathcal{V}} \sum_{p \in \mathcal{P}'^k} a_{ip}^k y_p^k = 1 \quad (\forall i \in \mathcal{W}) \quad (\text{A.69})$$

$$\sum_{p \in \mathcal{P}'^k} y_p^k = 1 \quad (\forall k \in \mathcal{V}) \quad (\text{A.70})$$

$$y_p^k \geq 0 \quad (\forall p \in \mathcal{P}'^k, \forall k \in \mathcal{V}) \quad (\text{A.71})$$

The dual variables of the RLMP are used to calculate the reduced cost of routes in the subproblem to be introduced below.

Subproblem

The main purpose of the subproblem is to find profitable routes by calculating the reduced cost for each route. Because the RLMP is a set partitioning problem, the subproblem essentially works with only elementary paths, which is called an elementary shortest path problem with resource constraints (ESPPRC) that is a special case of the SPPRC. After applying the DWD, l subproblems are defined. The non-negative dual variables of the RLMP, π and π_0^k , $k \in \mathcal{V}$, enable each subproblem to determine that

$$\bar{c}^k := \text{Minimize}\{(\mathbf{c}^{kT} - \boldsymbol{\pi}^T \mathbf{A}^k) \mathbf{x}^k - \pi_0^k\} \quad (\text{A.72})$$

which is the matrix expression of the objective function of the subproblem, Eq. (A.52).

The affine constraint, Eq. (A.70), can decompose into $\sum_{p \in \mathcal{P}^k} y_p^k \geq 1$ and $\sum_{p \in \mathcal{P}^k} y_p^k \leq 1$. Then, $\sum_{p \in \mathcal{P}^k} y_p^k \leq 1$ enables to have both an upper bound and a lower bound on f_{LMP} , Eq. (A.48), such that

$$f_{RLMP} + \sum_{k \in \mathcal{V}} \bar{c}^k \leq f_{LMP} \leq f_{RLMP} \quad (\text{A.73})$$

where $f_{RLMP} = \boldsymbol{\pi}^T \mathbf{b} + \sum_{k \in \mathcal{V}} \pi_0^k$ by the LP duality. In each iteration, f_{RLMP} can decrease up to the aggregate of the smallest reduced costs \bar{c}^k for each subproblem, $k \in \mathcal{V}$ [140]. Because of the condition, $f_{LMP} \leq f_{IMP}$, $f_{RLMP} + \sum_{k \in \mathcal{V}} \bar{c}^k$ is a lower bound on f_{IMP} . If $\bar{c}^k \geq 0$, there is no negative reduced cost column, which means there is no profitable route for vehicle k . At an optimal point, $\bar{c}^k = 0$ for all $k \in \mathcal{V}$, $f_{LMP} = f_{RLMP}$.

Stop criterion

If there is no profitable route for all vehicles, $\bar{c}^k \geq 0$ for all $k \in \mathcal{V}$, the iterative process terminates. Otherwise, the routes having a negative reduced cost are added to the RLMP as new columns of the constraints matrix. For instance, suppose the RLMP has p routes at the i -th iteration. After updating the master problem in the $(i + 1)$ -th iteration, it has p' routes,

	y_1	y_2	y_3	\cdots	y_p
w_1	a_{11}	a_{12}	a_{13}	\cdots	a_{1p}
w_2	a_{21}	a_{22}	a_{23}	\cdots	a_{2p}
w_3	a_{31}	a_{32}	a_{33}	\cdots	a_{3p}
\vdots	\vdots	\vdots	\vdots		\vdots
w_n	a_{n1}	a_{n2}	a_{n3}	\cdots	a_{np}

(a) The i -th iteration

	y_1	y_2	y_3	\cdots	y_p	y_{p+1}	\cdots	$y_{p'}$
w_1	a_{11}	a_{12}	a_{13}	\cdots	a_{1p}	$a_{1(p+1)}$	\cdots	$a_{1p'}$
w_2	a_{21}	a_{22}	a_{23}	\cdots	a_{2p}	$a_{2(p+1)}$	\cdots	$a_{2p'}$
w_3	a_{31}	a_{32}	a_{33}	\cdots	a_{3p}	$a_{3(p+1)}$	\cdots	$a_{3p'}$
\vdots	\vdots	\vdots	\vdots		\vdots	\vdots		\vdots
w_n	a_{n1}	a_{n2}	a_{n3}	\cdots	a_{np}	$a_{n(p+1)}$	\cdots	$a_{np'}$

(b) The $(i+1)$ -th iteration

Figure A.14: Shapes of constraint matrix at the i -th and the $(i+1)$ -th iterations of column generation.

$p' > p$, as described in Fig. A.14. After the iterative process terminates, all routes of the RLMP are transferred to the RIMP.

Label correcting algorithm and turning effects

In order to accelerate the column generation process, the label correcting algorithm [148] can be utilized to solve subproblems, which finds all routes with a negative reduced cost in the subproblem. The algorithm is a dynamic programming method that divides a given problem into simpler subproblems in a recursive manner. In general approaches, even if column generation works with route-based variables, arc-based variables needs to be used to reorganize routes because the list of waypoints visited can be obtained from a_{ip}^k and y_p^k , but the order of them cannot be traced without arc-based variables. When utilizing the label correcting algorithm, the RLMP can be solved based on just route-based variables by storing path information in each label. This algorithm can add turning penalties to cost coefficients of the RLMP and subproblems while the linearity of the RLMP remains [130].

In the label correcting algorithm, a label of vehicle k , $k \in \mathcal{V}$, at node z , $z \in \mathcal{N}$, can be defined as

$$\mathbf{L}_z^k = (\mathbf{R}_z^k, \mathbf{Q}_z^k, C_z^k), (\forall z \in \mathcal{N}, \forall k \in \mathcal{V}) \quad (\text{A.74})$$

where \mathbf{Q}_z^k is a ordered list of arcs the label moves, and C_z^k is a route cost along the label, and \mathbf{R}_z^k is resources used by vehicle k , $k \in \mathcal{V}$, from a depot to node z , $z \in \mathcal{N}$ which is given by

$$\mathbf{R}_z^k = (E_z^k, \mathbf{U}^k = [U_0^k, \dots, U_{n+1}^k], \sum_{z \in \mathcal{N}} U_z^k) \quad (\text{A.75})$$

where \mathbf{U}^k is a vector of unreachable nodes of vehicle k : if node i is already visited, $u_z^k = 1$. Otherwise, $u_z^k = 0$. In addition, E_z^k is energy used by vehicle k , $k \in \mathcal{V}$, from a depot to node z , $z \in \mathcal{N}$, which is given by

$$E_z^k = \sum_{(0,j) \in \mathbf{Q}_z^k} e_T^k t_T^k + \sum_{(i,j) \in \mathbf{Q}_z^k} e_F^k d_{ij}^k + \sum_{(i,h),(h,j) \in \mathbf{Q}_z^k} T_{ihj}^k + e_M^k t_h^k + \sum_{(i,n+1) \in \mathbf{Q}_z^k} e_L^k t_L^k \quad (\text{A.76})$$

where T_{ihj}^k is energy required to turn a vehicle k into a direction of arc (h, j) from a direc-

tion of arc (i, h) . Using energy per unit time required by vehicle k , $k \in \mathcal{V}$, to turn, e_R^k , T_{ihj}^k can be calculated by

$$T_{ihj}^k = e_R^k \frac{\theta_{ihj}^k}{r^k} \quad (\text{A.77})$$

where θ_{ihj}^k is the angle between arc (i, h) and arc (h, j) for vehicle k , and r^k is a turning rate of vehicle k .

The label correcting algorithm is working with a partial order of resources to accelerate run time. Only non-dominated labels are propagated to neighbor nodes. When $C_z^k = E_z^k$, the dominance rule can be described as shown in Algorithm 3. The main idea is that when there are labels which visit same nodes, only non-dominated labels are transmitted to next iterations. For the whole procedures of the label correcting algorithm and numerical examples, refer [130, 148].

Algorithm 3 Pseudo code for evaluating dominance

Input: $L_{z_1}^{k_1}, L_{z_2}^{k_2}$

Output: *True* if $L_{z_1}^{k_1}$ is dominated by $L_{z_2}^{k_2}$, otherwise *False*

if $(k_1 \neq k_2)$ or $(\sum_{z \in \mathcal{N}} U_{z_1}^{k_1} \neq \sum_{z \in \mathcal{N}} U_{z_2}^{k_2})$ **then**
 return *False*

else

for i in \mathcal{N} **do**

if $U_i^{k_1} \neq U_i^{k_2}$ **then**

return *False*

end if

end for

if $E_{z_1}^{k_1} \leq E_{z_2}^{k_2}$ **then**

return *False*

else

return *True*

end if

end if

A.3 Multi-UAS Path-Planning for a Large-scale Disjoint Disaster Management: Mathematical Formulations³

A.3.1 Introduction of Disaster Management Missions

Recently, unmanned aircraft systems (UAS) have become a promising technology to execute high-risk missions such as monitoring natural disasters and nuclear explosions because of an advanced compact and energy-efficient system. Particularly, the use of UAS is suitable to monitor wildfire that requires fast-response for predicting the future direction of motion of a wildfire and its impact. There are two typical missions for wildfire monitoring: high-altitude disaster monitoring (HADM) and low-altitude fire perimeter monitoring (LAFPM) [150, 151]. The high-altitude disaster monitoring collects comprehensive information in an entire wildfire area to analyze its intensity and damages. The low-altitude fire perimeter monitoring observes the rate of its spread in a real-time manner.

The mission management research for both wildfire monitoring missions recently has actively been researched. In the LAFPM mission, Casbeer et al. [152] introduced the decentralized multiple-UAV concept for tracking the propagation of large forest fires that minimizes the information latency and the frequency of update. Phan and Liu [153] proposed the concept of a cooperative forest fire observation with three-layered hierarchical vehicle platforms with a blimp, UAVs, and rovers. For the coordination of the observation task, the paper applies an integer linear programming. Pham et al. [154] suggested the distributed wildfire monitoring management algorithm using a potential field technique for the observation of a dynamic fire trend. Bailon-Ruiz et al. [155] presented the planning algorithm based on a generic variable neighborhood search method to observe spreading wildfires.

The HADM mission is associated with large-scale tasks such as the Western States Fire Mission which provides 24-hour coverage of fires [150, 151]. The HADM mission plan-

³This section is accepted for publication [149].

ning resembles a coverage path planning (CPP) to scan multiple disjoint areas, which builds an optimal coverage route to scan all the disjoint areas [156]. A limitation of these approaches cannot handle large disjoint areas since the optimization framework, a travelling salesman problem (TSP), only considers a single UAV operation. To resolve this limitation, the optimization framework for a multi-UAVs path-planning should be considered to completed the large disjoint areas.

This paper focuses on the HADM problem and introduces a novel multi-UAS path-planning algorithm to monitor multiple and large-scale wildfires. To capture a practical UAS operation, it considers multiple ground stations where UAS are potentially deployed. For the optimal path-planning, we combine an exact CPP method and a multi-depot vehicle routing problem (MDVRP) model that generates computationally efficient monitoring solution through decomposing simpler problems.

A.3.2 A Framework for a UAS-based Wildfire Scanning Mission

Solving VRP is an NP-hard problem that exponentially increases computation time as its problem size increases. In a large-scale wildfire scanning mission, if it has multiple disjoint wildfires with large area, it is easily not solvable because of the complexity. To overcome this computational issue, we proposes a new method that makes more computationally favorable through dividing into two subproblems as shown in Fig. A.15: Coverage path-planning and operational planning. The coverage path-planning simply generates actual flight coverage path of each wildfire. The operational planning builds the optimal path that defines the best sequence visiting all the wildfires. This approach can significantly reduce the problem size of the optimization model compared to solving the entire problem at once.

Coverage path planning for Dubins vehicles

The coverage path planning (CPP) methods for aerial imaging missions have been proposed diverse approaches [135, 157, 131, 130]. The notable CPP methods can be cat-

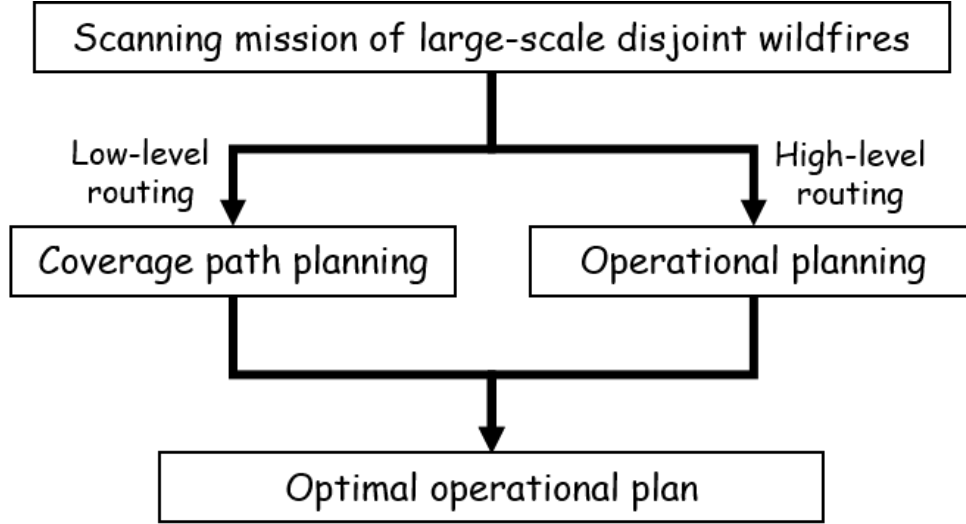


Figure A.15: Notional decomposition of the large-scale scanning mission

egorized into three groups: a classical exact method, a wavefront-based method, and a vehicle-routing-based method [130]. The classical exact method creates a Back and Forth Pattern (BFP) path to cover a scanning area, the wavefront-based method applies a wave propagation function that assigns a value of each grid cell and generates a route based on those values. The vehicle-routing-based method formulates an optimization problem that can handle multi-UAV operations to scan a scanning area.

For a HADM mission, high-altitude long-endurance (HALE) UAVs are preferred [151] because typical HADM missions require a long endurance vehicle. This paper assumes that a UAV has a long-endurance platform that can fully observe at least one target area. This assumption allows to create a coverage path of a single UAV for each area by a classical exact method which is faster than other methods because the BFP can be directly utilized without requiring additional computation power to build a trajectory. Thus, this paper utilizes a classical exact method to build a scanning path for each wildfire. The readers interested in the CPP methods may refer the extensive surveys of Galceran and Carreras [158] and Khan et al. [159].

In the CPP problems, minimizing the number of turns is critical to reducing the total mission time or the total energy [136]. The classical exact methods of the CPP decrease

the number of turns on their routes by creating the BFP path along the optimal line-sweep direction of the area of interest (AOI). The line-sweep direction of a convex area is determined by a diameter function which is its height in a reference coordinate system calculated by rolling the area. The line-sweep direction having a minimum of the diameter function of a convex area guarantees the minimum number of turns on the BFP route along the direction [136].

As a notional example, let us consider a convex hull modeled by an AOI. When establishing grid cells, a reference coordinate system of the whole AOI could be utilized for all target areas. Then, every subarea has the same coordinate system. Figure A.16a shows a coverage path built by grid cells defined in a reference coordinate system consisting of a horizontal and a vertical lines. When employing an optimal line-sweep direction for each subarea, a local coordinate system consisting of a line of the direction and a perpendicular line should be created for each subarea. Then, a coverage path built by grid cells defined in the local coordinate system has the number of turns less than or equal to a coverage path in a reference coordinate system as illustrated in Fig. A.16b. Note that if a convex hull is not symmetric in terms of its line-sweep direction, the total length of a coverage path depends on its initial scan direction while the paths have the same number of turns. For these cases, we select the shorter path to scan the convex hull created by an AOI.

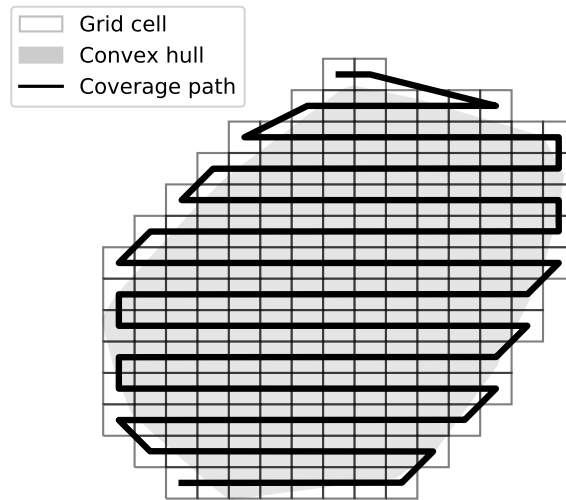
The BFP path created by classic CPP methods may include sharp turns that occur at a point. To address more realistic turn performance of fixed wing UAVs, the UAVs are considered as Dubins vehicles having a kinematic model as follows:

$$\dot{x} = V \cos \theta \quad (\text{A.78})$$

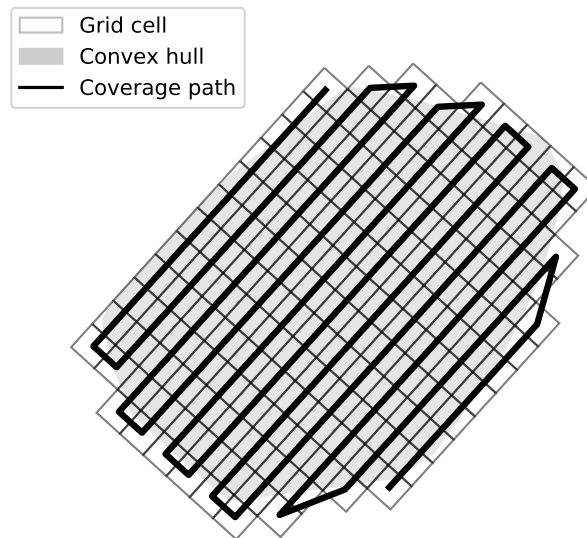
$$\dot{y} = V \sin \theta \quad (\text{A.79})$$

$$\dot{\theta} = u, \quad (\text{A.80})$$

where (x, y) is its position, V is the speed, θ is the heading angle, and u is the minimum

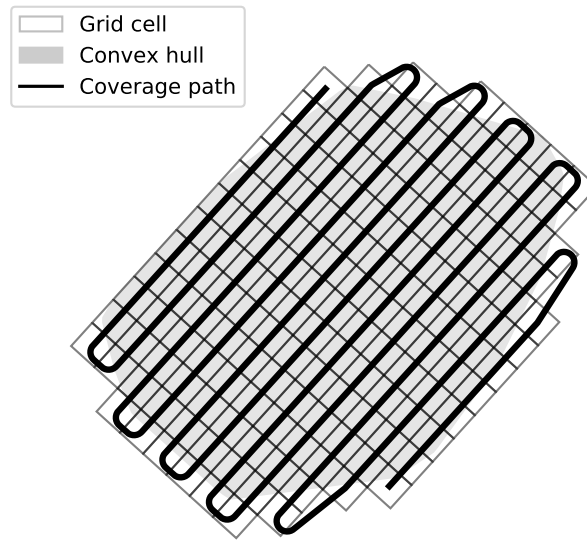


(a) Coverage path without considering a line-sweep direction (26 turns)

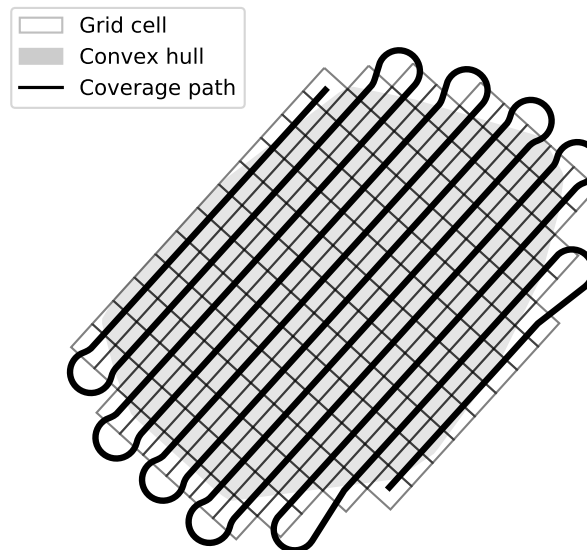


(b) Coverage path along an optimal line-sweep direction (20 turns)

Figure A.16: The effect of a line-sweep direction on creating a coverage path



(a) Turns of a Dubins vehicle having enough turning performance



(b) Turns of a Dubins vehicle having not enough turning performance

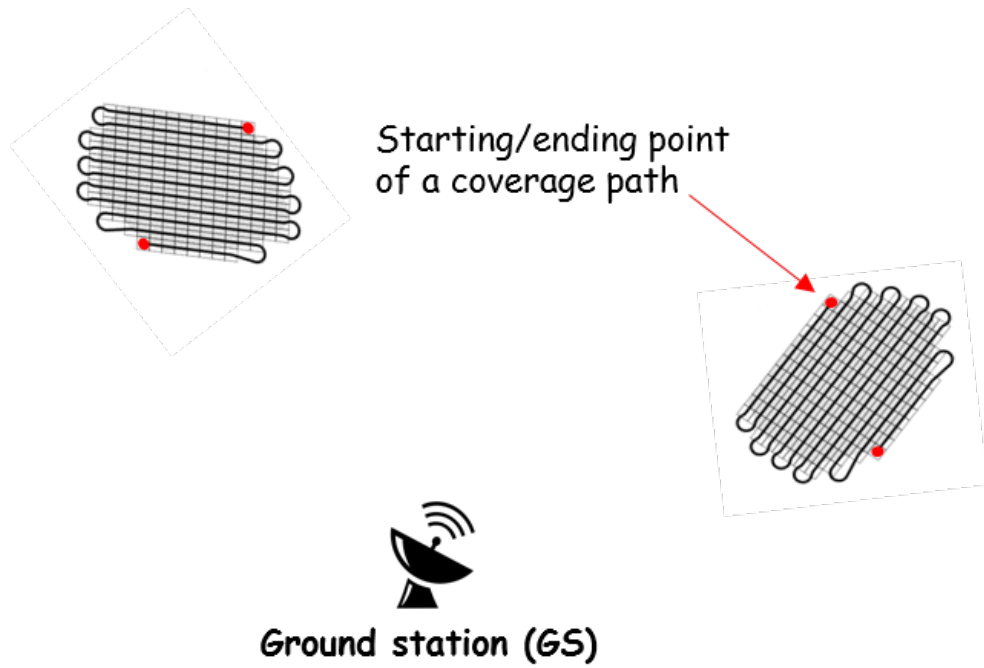
Figure A.17: The effect of turning performance on turns of a Dubins' vehicle

turning radius or the maximum curvature of the vehicle. Dubins [160] proves that the optimal path between two points constrained by vehicle's heading angle and minimum turning radius has at most CLC or CCC patterns, where C is a turn with its minimum turning radius, and L is a straight line. Johnson [161] and Boissonnat et al. [162] show the same result with control theory.

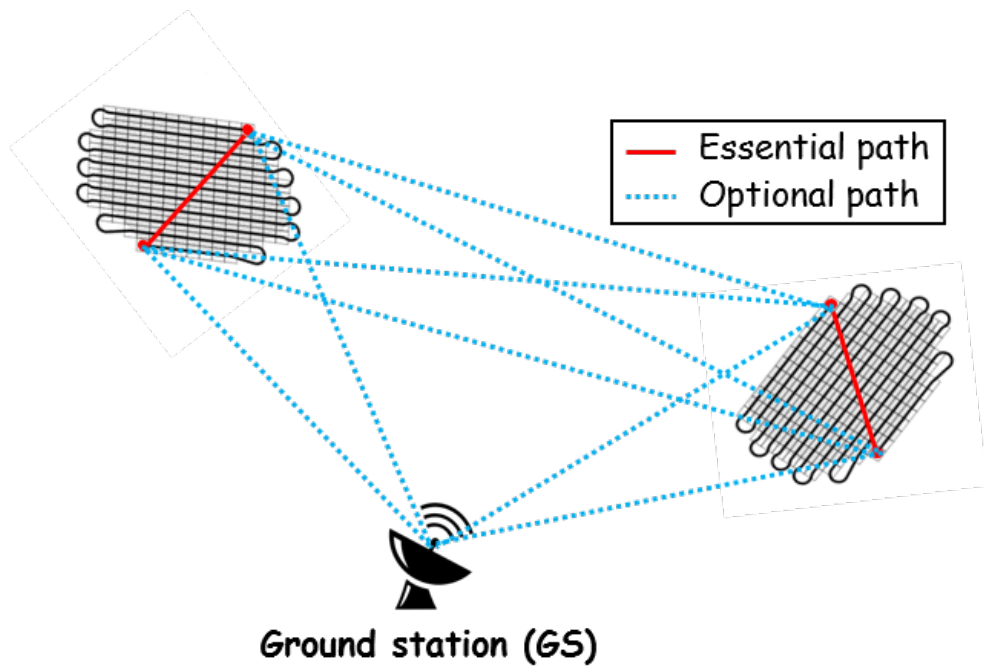
To address the properties of Dubins vehicles, Dubins paths are built for each pair of two turning points where the scanning direction is changed. The turn performance of UAVs has an effects on a length of curve for turning as illustrated in Fig. A.17. If the turn performance of UAVs is not enough to turn between two waypoints, the turn motion requires extra flight time and energy. In this case, reducing the number of turns could be more significant to minimize flight time and energy.

Formulations for operational planning

The operational planning is conducted with a simplified flight network that does not consider all the complex low-level flight network. The new network for the operational planning includes two points, mission start/end locations resulting from the CPP result, for each target area as shown in Fig. A.18a. The new flight network simply connects between a GS and a mission start/end to generate initial potential paths as illustrated in Fig. A.18b. In this way, the complexity of initial flight network can significantly reduce; the complexity relies on the topology of the flight network.

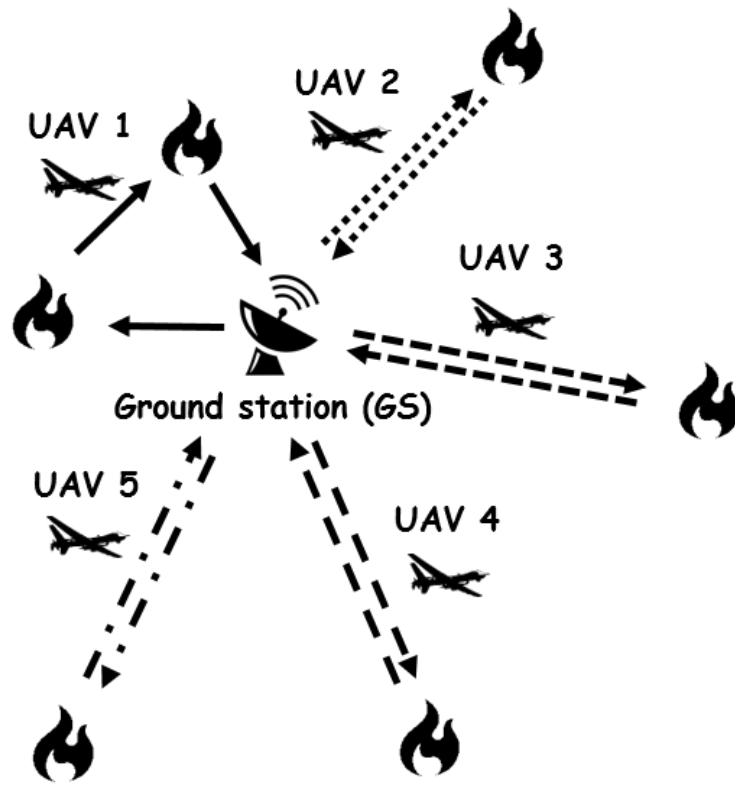


(a) Step 1: finding starting/ending point of each coverage path

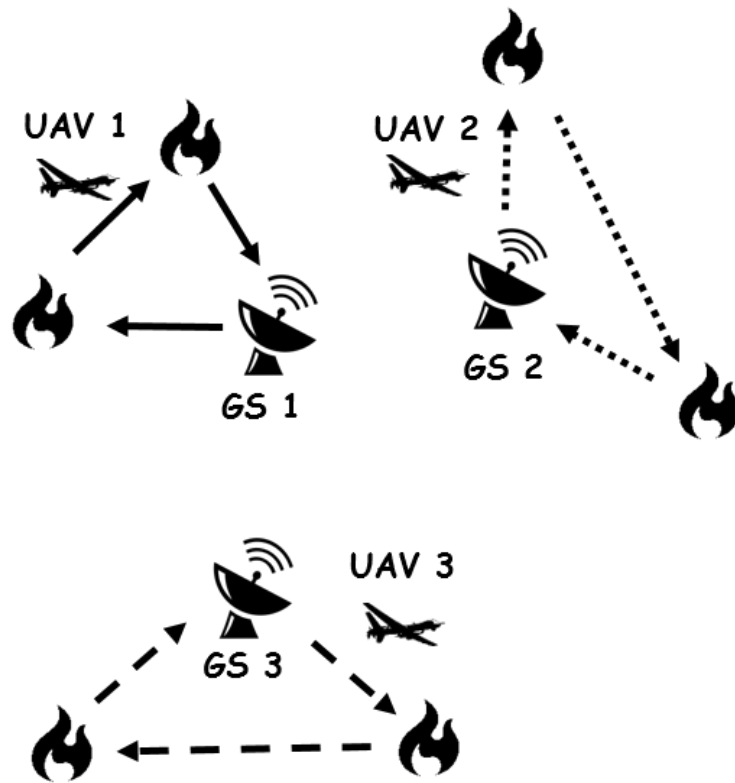


(b) Step 2: creating a flight network as an input of the MDVRP model

Figure A.18: The modeling of mission areas by a graph



(a) ConOps for the VRP



(b) ConOps for the MDVRP

Figure A.19: The effect of vehicle routing formulations on a ConOps

$$\text{Minimize} \quad \sum_{i \in \mathcal{N}_{NA}} \sum_{j \in \mathcal{N}_{ND}} \sum_{d \in \mathcal{N}_D} \sum_{k \in \mathcal{V}^d} \mathcal{D}_{ij}^{dk} x_{ij}^{dk} \quad (\text{A.81})$$

Subject to

$$\sum_{j \in \mathcal{N}_{ND}} \sum_{d \in \mathcal{N}_D} \sum_{k \in \mathcal{V}^d} x_{ij}^{dk} = 1 \quad (\forall i \in \mathcal{N}_M) \quad (\text{A.82})$$

$$\sum_{i \in \mathcal{N}_D} \sum_{j \in \mathcal{N}_{ND}} x_{ij}^{dk} = 1 \quad (\forall d \in \mathcal{N}_D, \forall k \in \mathcal{V}^d) \quad (\text{A.83})$$

$$\sum_{i \in \mathcal{N}_{NA}} \sum_{j \in \mathcal{N}_A} x_{ij}^{dk} = 1 \quad (\forall d \in \mathcal{N}_D, \forall k \in \mathcal{V}^d) \quad (\text{A.84})$$

$$\sum_{j \in \mathcal{N}_{ND}} \sum_{k \in \mathcal{V}^d} x_{dj}^{dk} = |\mathcal{V}^d| \quad (\forall d \in \mathcal{N}_D) \quad (\text{A.85})$$

$$\sum_{j \in \mathcal{N}_{ND}} x_{dj}^{dk} - \sum_{i \in \mathcal{N}_{NA}} x_{i(d+|D|)}^{dk} = 0 \quad (\forall d \in \mathcal{N}_D, \forall k \in \mathcal{V}^d) \quad (\text{A.86})$$

$$\sum_{i \in \mathcal{N}_{NA}} x_{ih}^{dk} - \sum_{j \in \mathcal{N}_{ND}} x_{hj}^{dk} = 0 \quad (\forall h \in \mathcal{N}_M, \forall d \in \mathcal{N}_D, \forall k \in \mathcal{V}^d) \quad (\text{A.87})$$

$$\sum_{d \in \mathcal{N}_D} \sum_{k \in \mathcal{V}^d} x_{ij}^{dk} + \sum_{d \in \mathcal{N}_D} \sum_{k \in \mathcal{V}^d} x_{ji}^{dk} = 1 \quad (\forall (i, j) \in \mathcal{A}_{PM}) \quad (\text{A.88})$$

$$\sum_{j \in \mathcal{N}_{ND}} y_{ij}^{dk} - \sum_{j \in \mathcal{N}_{NA}} y_{ji}^{dk} - \sum_{j \in \mathcal{N}_{ND}} \mathcal{D}_{ij}^{dk} x_{ij}^{dk} = 0 \quad (\forall i \in \mathcal{N}_M, \forall d \in \mathcal{N}_D, \forall k \in \mathcal{V}^d) \quad (\text{A.89})$$

$$y_{dj}^{dk} = \mathcal{D}_{dj}^{dk} x_{dj}^{dk} \quad (\forall j \in \mathcal{N}_{ND}, \forall d \in \mathcal{N}_D, \forall k \in \mathcal{V}^d) \quad (\text{A.90})$$

$$y_{ij}^{dk} \geq (\mathcal{D}_{di}^{dk} + \mathcal{D}_{ij}^{dk}) x_{ij}^{dk} \quad (\forall i \in \mathcal{N}_{NA}, \forall j \in \mathcal{N}_{ND}, \forall d \in \mathcal{N}_D, \forall k \in \mathcal{V}^d) \quad (\text{A.91})$$

$$y_{ij}^{dk} \leq (R^{dk} - \mathcal{D}_{j(d+|D|)}^{dk}) x_{ij}^{dk} \quad (\forall i \in \mathcal{N}_{NA}, \forall j \in \mathcal{N}_{ND}, \forall d \in \mathcal{N}_D, \forall k \in \mathcal{V}^d) \quad (\text{A.92})$$

$$y_{i(d+|D|)}^{dk} x_{i(d+|D|)}^{dk} \leq R^{dk} x_{i(d+|D|)}^{dk} \quad (\forall i \in \mathcal{N}_{NA}, \forall d \in \mathcal{N}_D, \forall k \in \mathcal{V}^d). \quad (\text{A.93})$$

The operational planning should consider the concept of operations and the vehicle constraints from UAVs characteristics. To introduce the mathematical formulation of the operational planning, we assume two conceptual examples of a 6-wildfire monitoring mission as shown in Fig. A.19: single-GS and multi-GS. In the single-GS scenario with a single UAV, it can be typically modeled by the traveling salesman problem (TSP). In the single-GS scenario with multiple UAVs, a vehicle routing problem (VRP) is typically applied, which is depicted in Fig. A.19a. The VRP is a generalized form of the TSP. In the multi-GS scenario with multiple UAVs shown in Fig. A.19b, we can implement the multi-depot vehicle routing problem (MDVRP) that optimizes the deployment locations of UAVs and their routes. The MDVRP can have a benefit of having operating cost because UAVs can be deployed the GS closed to the wildfire, and the total mission time and the usage of the number of UAVs can significantly reduce. Because of these benefits, this paper applies the MDVRP to address the wildfire management problem.

To address a generalized delivery system having multiple depots and vehicles, the MDVRP is formulated as a variant of the VRP [163, 164]. In UAS-based applications, the formulations of the MDVRP are used to consider several GSs, which are potential UAV depots, in a post-earthquake assessment mission [165]. The readers interested in the MDVRP could refer the comprehensive survey conducted by Montoya-Torres et al. [166].

To address the MDVRP formulation, we extend the optimization model based on the distance-constrained VRP (DVRP) proposed by Choi et al. [130], which has the generalized formulation of Kara's VRP model [137] to obtain the minimum required number of vehicles. However, this can deal with multiple UASs with a single GS only. The MDVRP as a variant of the VRP considering multiple GSs can be described with graph theory. Mission areas are depicted by a graph, $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, and each depot and mission point is modeled by a node. Sets, \mathcal{N}_D and \mathcal{N}_M , represent all depots and mission points respectively. In our MDVRP model, an artificial depot node is added for each depot node. A set \mathcal{N}_A indicates all artificial depots. Note that a physical depot is described with two nodes; a depot node and

an artificial depot node. The depot node and the artificial depot node are used to represent a starting point and an ending point in a route. For mathematical simplification, we define $|\cdot|$ as the number of elements of a set. For instance, a graph \mathcal{G} includes $|M|$ number of mission points, $M = \{0, 1, \dots, |M| - 1\}$, $|D|$ number of depots, $D = \{0, 1, \dots, |D| - 1\}$, and $|A|$ number of artificial depots, $A = \{0, 1, \dots, |A| - 1\} = \{0, 1, \dots, |D| - 1\}$. The set of nodes \mathcal{N} and its subsets are defined such that

$$\begin{aligned}
\mathcal{N}_M &= \{0, 1, \dots, |M| - 1\} \\
\mathcal{N}_D &= \{|M|, |M| + 1, \dots, |M| + |D| - 1\} \\
\mathcal{N}_A &= \{|M| + |D|, |M| + |D| + 1, \dots, |M| + 2|D| - 1\} \\
\mathcal{N} &= \mathcal{N}_M \cup \mathcal{N}_D \cup \mathcal{N}_A \\
&= \{0, 1, \dots, |M| + 2|D| - 1\} = \{0, 1, \dots, |\mathcal{N}| - 1\} \\
\mathcal{N}_{ND} &= \mathcal{N} - \mathcal{N}_D = \mathcal{N}_M \cup \mathcal{N}_A \\
\mathcal{N}_{NA} &= \mathcal{N} - \mathcal{N}_A = \mathcal{N}_M \cup \mathcal{N}_D.
\end{aligned}$$

An arc presents an edge where a vehicle travels from point A to point B , and the set of arcs \mathcal{A} in the graph \mathcal{G} can be expressed by $\mathcal{A} = \{(i, j) : \forall i \in \mathcal{N}_{NA}, \forall j \in \mathcal{N}_{ND}, \text{ and } i \neq j\}$. To connect the mission start/end locations in the simplified flight network, a set $\mathcal{A}_{PM} = \{(i, i + 1) : \forall i \text{ is an even number of } \mathcal{N}_M\}$ is defined, which corresponds the pair of two locations, mission start/end positions. A set $\mathcal{V} = \bigcup_{d=0}^{|D|-1} \mathcal{V}^d$ indicates a fleet of vehicles and \mathcal{V}^d represents vehicles available at a depot d , $\forall d \in D$, such that $\mathcal{V}^d = \{0, 1, \dots, |\mathcal{V}^d| - 1\}$, and \mathcal{V}^{dk} is each vehicle at a depot d , $\forall d \in D, \forall k \in \mathcal{V}^d$. In the equations, a superscript is related to a depot or a vehicle, and subscripts are associated with an arc. Each vehicle \mathcal{V}^{dk} has its maximum range R^{dk} , where $\forall d \in D, \forall k \in \mathcal{V}^d$. The distance from a node i to another node j traveled by a vehicle \mathcal{V}^{dk} is defined as $\mathcal{D}_{ij}^{dk} (\forall (i, j) \in \mathcal{A}, \forall d \in D, \forall k \in \mathcal{V}^d)$. The mathematical formulation has two types of design variables: vehicle movements, x_{ij}^{dk} , and

flight distances, y_{ij}^{dk} , where $\forall(i, j) \in \mathcal{A}, \forall d \in D, \forall k \in \mathcal{V}^d$. This 4-index-based formulation that has been used by Salhi et al. [167] includes starting/ending nodes, originated depots, and a vehicle type. The vehicle movement design variables, x_{ij}^{dk} , are integer variables. The flight distance variables, y_{ij}^{dk} , are real variables having the total flight distance of a vehicle \mathcal{V}^{dk} , which flies from a node i to another node j , from a depot d to the node j along its route.

Based on these definitions, an optimization model for multi-depot vehicle routing problems for a UAS-based wildfire monitoring mission in disjoint areas can be written by Eq. (A.81)-(A.93). The objective function, Eq. (A.81), is the total flight distance traveled by all vehicles. The constraints, Eq. (A.82)-(A.93), describe the ConOps of a UAS; each mission point is visited by a UAV exactly once, Eq. (A.82). Each vehicle departs from a depot, Eq. (A.83), and arrive at a depot, Eq. (A.84). The number of available vehicles is limited for each depot, Eq. (A.85). Each vehicle that departs from a depot should return to the same depot, Eq. (A.86). If a vehicle visits a mission point, it should move to another mission point or a depot, Eq. (A.87). Eq. (A.88) presents that the CPP mission should be conducted. When a vehicle moves from a point to another point, its total flight distance increases as much as it flies, Eq. (A.89), which provides a subtour elimination. The initial flight distance of each vehicle is set by Eq. (A.90). The lower and upper bounds of flight distance given a vehicle at a specific mission point is written by Eq. (A.91) and Eq. (A.92) respectively. The upper bound of flight distance of a vehicle returning to a depot is described by Eq. (A.93). The overall process of the developed framework for multi-UAS path-planning for a large-scale disjoint disaster management is summarized in Algorithm 4.

Algorithm 4 Framework for multi-UAS path-planning for a large-scale disjoint disaster management

Input: wildfire information, vehicle/GS parameters

Output: optimal routes for each vehicle

PART 1: coverage path planning - [Section A.3.2]

for each wildfire in wildfires **do**

Find the convex hull of an area of a wildfire

Find an optimal line-sweep direction

Create grid cells based on its local coordinate system

Build two coverage paths having different scan directions

Update Dubins paths for each turn

Select the shorter path

Extract starting/ending point of the path

end for

PART 2: operational planning - [Section A.3.2]

Select candidate GSs

Build a flight network based on Dubins paths as an input for the MDVRP model

Solve the MDVRP - [Eq. (A.81)-(A.93)]

Combine the solution of the MDVRP model with coverage paths

return Optimal routes for each UAV

APPENDIX B

REGULATORY REQUIREMENTS FOR SUAS

According to The Code of Federal Regulations (14 CFR Part 1¹), public and civil aircrafts are defined as follows:

public aircraft : An aircraft used only for the United States Government or owned by the Government or owned and operated by the government of a State.

civil aircraft : It means aircraft other than public aircraft.

Note that if public aircraft is used for commercial purposes, its status is changed to that of civil aircraft. That is, the status of aircraft depends on the type of operation it is conducting at that time. The reason why categorizing aircraft into two groups is that the regulatory requirements of them are different significantly due to the fact that public aircraft are legally exempt from most types of FAA regulation such as civil airworthiness and airman certification [168].

On the other hand, the FAA categorizes operators into four groups: remote pilots holding a Part certificate, public UAS operators, model aircraft operators, and those holding 333 exemptions [169]. The most of operators for commercial operations are fallen into Part 107 in Table B.1 remote pilots category, whereas hobbyists are included in model aircraft operators [7]. As shown in Table B.2, these operators are affected by different regulatory requirements.

¹<https://www.ecfr.gov/cgi-bin/text-idx?rgn=div5&node=14:1.0.1.1.1> (accessed September 16, 2017)



Figure B.1: Airspace classification²

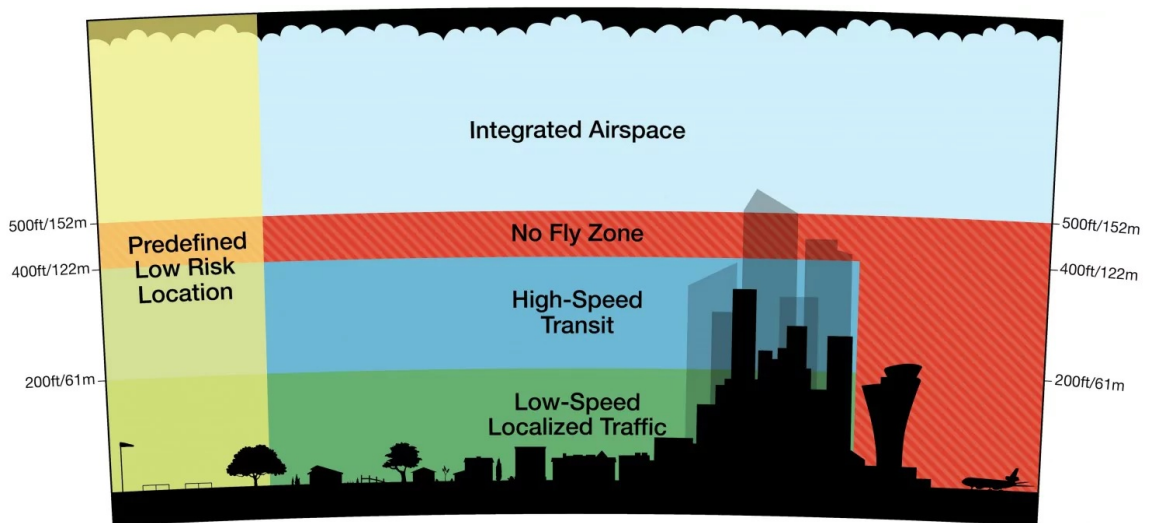


Figure B.2: Amazon's plan to use airspace³

²https://www.faa.gov/gslac/ALC/course_content.aspx?CID=42&SID=505&preview=true (accessed 2/19/2019)

³[https://utm.arc.nasa.gov/docs/Amazon_Revising%20the%20Airspace%20Model%20for%20the%20Safe%20Integration%20of%20sUAS\[6\].pdf](https://utm.arc.nasa.gov/docs/Amazon_Revising%20the%20Airspace%20Model%20for%20the%20Safe%20Integration%20of%20sUAS%20[6].pdf) (accessed 2/19/2019)

Part 107 also includes airspace regulation for sUAS. As seen in Figure B.1, airspace is divided into six Classes: from Class A to Class G. The operation of sUAS is not permitted in Class A. For operation in Class B, C, D, and E needs a permission from air traffic control (ATC), whereas operation in Class G does not require a permission from ATC. According to FAA⁴, the Classes are defined as follows:

Class A : airspace from 18,000 feet MSL up to and including FL 600

Class B : airspace from the surface to 10,000 feet MSL surrounding the nation's busiest airports in terms of IFR operations or passenger enplanements

Class C : airspace from the surface to 4,000 feet above the airport elevation (charted in MSL) surrounding those airports that have an operational control tower

Class D : airspace from the surface to 2,500 above the airport elevation (charted in MSL) surrounding those airports that have an operational control tower

Class E : if the airspace is not Class A, B, C, or D, and is controlled airspace it is Class E airspace

⁴https://www.faa.gov/gslac/ALC/course_content.aspx?cID=42&sID=505&preview=true (accessed 2/19/2019)

Table B.1: Summary of small unmanned aircraft rule (Part 107) [170]

Operational Limitations	<ul style="list-style-type: none"> • Unmanned aircraft must weigh less than 55 lbs. (25 kg). • Visual line-of-sight (VLOS) only; the unmanned aircraft must remain within VLOS of the remote pilot in command and the person manipulating the flight controls of the small UAS. Alternatively, the unmanned aircraft must remain within VLOS of the visual observer. • At all times the small unmanned aircraft must remain close enough to the remote pilot in command and the person manipulating the flight controls of the small UAS for those people to be capable of seeing the aircraft with vision unaided by any device other than corrective lenses. • Small unmanned aircraft may not operate over any persons not directly participating in the operation, not under a covered structure, and not inside a covered stationary vehicle. • Daylight-only operations, or civil twilight (30 minutes before official sunrise to 30 minutes after official sunset, local time) with appropriate anti-collision lighting. • Must yield right of way to other aircraft. • May use visual observer (VO) but not required. • First-person view camera cannot satisfy “see-and-avoid” requirement but can be used as long as requirement is satisfied in other ways. • Maximum groundspeed of 100 mph (87 knots).
-------------------------	--

- Maximum altitude of 400 feet above ground level (AGL) or, if higher than 400 feet AGL, remain within 400 feet of a structure.
- Minimum weather visibility of 3 miles from control station.
- Operations in Class B, C, D and E airspace are allowed with the required ATC permission.
- Operations in Class G airspace are allowed without ATC permission.
- No person may act as a remote pilot in command or VO for more than one unmanned aircraft operation at one time.
- No operations from a moving aircraft.
- No operations from a moving vehicle unless the operation is over a sparsely populated area.
- No careless or reckless operations.
- No carriage of hazardous materials.
- Requires preflight inspection by the remote pilot in command.
- A person may not operate a small unmanned aircraft if he or she knows or has reason to know of any physical or mental condition that would interfere with the safe operation of a small UAS.
- Foreign-registered small unmanned aircraft are allowed to operate under part 107 if they satisfy the requirements of part 375.

	<ul style="list-style-type: none"> • External load operations are allowed if the object being carried by the unmanned aircraft is securely attached and does not adversely affect the flight characteristics or controllability of the aircraft. • Transportation of property for compensation or hire allowed provided that - <ul style="list-style-type: none"> ◦ The aircraft, including its attached systems, payload and cargo weigh less than 55 pounds total; ◦ The flight is conducted within visual line of sight and not from a moving vehicle or aircraft; and ◦ The flight occurs wholly within the bounds of a State and does not involve transport between (1) Hawaii and another place in Hawaii through airspace outside Hawaii; (2) the District of Columbia and another place in the District of Columbia; or (3) a territory or possession of the United States and another place in the same territory or possession. • Most of the restrictions discussed above are waivable if the applicant demonstrates that his or her operation can safely be conducted under the terms of a certificate of waiver.
Remote Pilot in Command Certification and Responsibilities	<ul style="list-style-type: none"> • Establishes a remote pilot in command position. • A person operating a small UAS must either hold a remote pilot airman certificate with a small UAS rating or be under the direct supervision of a person who does hold a remote pilot certificate (remote pilot in command).

- To qualify for a remote pilot certificate, a person must:
 - Demonstrate aeronautical knowledge by either:
 - ▶ Passing an initial aeronautical knowledge test at an FAA-approved knowledge testing center; or
 - ▶ Hold a part 61 pilot certificate other than student pilot, complete a flight review within the previous 24 months, and complete a small UAS online training course provided by the FAA.
 - Be vetted by the Transportation Security Administration.
 - Be at least 16 years old.
- Part 61 pilot certificate holders may obtain a temporary remote pilot certificate immediately upon submission of their application for a permanent certificate. Other applicants will obtain a temporary remote pilot certificate upon successful completion of TSA security vetting. The FAA anticipates that it will be able to issue a temporary remote pilot certificate within 10 business days after receiving a completed remote pilot certificate application.
- Until international standards are developed, foreign-certificated UAS pilots will be required to obtain an FAA-issued remote pilot certificate with a small UAS rating.
- A remote pilot in command must:
 - Make available to the FAA, upon request, the small UAS for inspection or testing, and any associated documents/records required to be kept under the rule.

	<ul style="list-style-type: none"> ○ Report to the FAA within 10 days of any operation that results in at least serious injury, loss of consciousness, or property damage of at least \$500. ○ Conduct a preflight inspection, to include specific aircraft and control station systems checks, to ensure the small UAS is in a condition for safe operation. ○ Ensure that the small unmanned aircraft complies with the existing registration requirements specified in § 91.203(a)(2). ● A remote pilot in command may deviate from the requirements of this rule in response to an in-flight emergency.
Aircraft Requirements	<ul style="list-style-type: none"> ● FAA airworthiness certification is not required. However, the remote pilot in command must conduct a preflight check of the small UAS to ensure that it is in a condition for safe operation.
Model Aircraft	<ul style="list-style-type: none"> ● Part 107 does not apply to model aircraft that satisfy all of the criteria specified in section 336 of Public Law 112-95. ● The rule codifies the FAA's enforcement authority in part 101 by prohibiting model aircraft operators from endangering the safety of the NAS.

Table B.2: Comparison of model aircraft, section 333 exemptions, and proposed regulations under the small UAS notice of proposed rulemaking [7]

	Special rule for model aircraft and related guidelines	Section 333 exemptions covered under the blanket certificate of authorization	Small UAS notice of proposed rule-making
Aircraft weight	Less than 55 pounds or certified through a design, construction, inspection, flight test, and operational safety program administered by a community-based organization	Less than 55 pounds	Less than 55 pounds
Maximum aircraft speed	Not specified	Not specified	100 miles per hour (87 knots)
Maximum altitude	400 feet	200 feet	500 feet

Operator qualifications	None, although some FAA guidance material encourages operators to take lessons and learn to fly safely	A current pilot with an FAA airman certificate and either a valid FAA medical certificate or a U.S. driver's license	FAA unmanned aircraft operator certificate with small UAS rating requiring aeronautical knowledge tests every 24 months Transportation Security Administration terrorism threat assessment
Aircraft registration, and certification, and determination of airworthiness	Except for vehicles weighing less than 250 grams (0.55 pounds), the operator must register with FAA using an online form for small unmanned aircraft, and the assigned registration number must be affixed to the model aircraft	UAS must be registered with the FAA using a paper registration process prior to applying for the Section 333 exemption. After March 31, 2016, applicants will be able to register using the online registration form	Registration required and registration number must be displayed in standard size or as large as possible No FAA airworthiness certification required, but operator must properly maintain the UAS and carry out pre-flight inspections to ensure it is safe to operate

<p>Airspace and operational limitations</p>	<p>Within visual line-of-sight</p> <p>Clear of stadiums and people</p> <p>Not within five miles of an airport without prior notification</p>	<p>Within visual line-of-sight</p> <p>From sunrise to sunset with visibility of three miles or more</p> <p>At least five nautical miles (NM) from an airport with an operating control tower, three NM from an airport with a published instrument procedure but no operational tower, or two NM from a heliport with a published instrument procedure or an airport with no published instrument procedures</p>	<p>Within visual line-of-sight (may use an observer)</p> <p>May not fly over people not directly involved in flight operations</p> <p>A first-person view camera is permitted, but cannot be used to operate beyond visual line of sight or to satisfy see-and-avoid requirements</p>
---	--	--	---

REFERENCES

- [1] U.S. Department of Transportation John A. Volpe National Transportation Systems Center, “Unmanned aircraft system (UAS) service demand 2015 - 2035,” United States Air Force Aerospace Management Systems Division, Air Traffic Systems Branch (AFLCMC/HBAG), Tech. Rep., 2013.
- [2] M. S. Francis, “Unmanned air systems: Challenge and opportunity,” *Journal of Aircraft*, vol. 49, no. 6, pp. 1652–1665, 2012.
- [3] S. G. Gupta, M. M. Ghonge, and P. M. Jawandhiya, “Review of unmanned aircraft systems (UAS),” *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 2, no. 4, pp. 1646–1658, 2013.
- [4] “The DoD dictionary,” http://www.dtic.mil/doctrine/dod_dictionary/ (retrieved on 9/05/2017).
- [5] The Department of Defense (DOD). (2007). Unmanned systems roadmap 2007-2032.
- [6] A. C. Watts, V. G. Ambrosia, and E. A. Hinkley, “Unmanned aircraft systems in remote sensing and scientific research: Classification and considerations of use,” *Remote Sensing*, vol. 4, pp. 1671–1692, 2012.
- [7] B. Elias. (2016). Unmanned aircraft operations in domestic airspace: U.S. policy perspectives and the regulatory landscape, Congressional Research Service.
- [8] M. Narkus-Kramer, “Future demand and benefits for small-autonomous unmanned aerial systems package delivery,” in *17th AIAA Aviation Technology, Integration, and Operations Conference*, Denver, Colorado, 2017.
- [9] D. Locascio, M. Levy, K. Ravikumar, S. Briceno, B. German, and D. Mavris, “Evaluation of concepts of operations for on-demand package delivery by small unmanned aerial systems,” in *16th AIAA Aviation Technology, Integration, and Operations Conference*, Washington, D.C., 2016.
- [10] N. Mathew, S. L. Smith, and S. L. Waslander, “Planning paths for package delivery in heterogeneous multirobot teams,” *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1298–1308, 2015.

- [11] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 70–85, 2017.
- [12] "Amazon Prime drone," <https://www.amazon.com/Amazon-Prime-Air/b?node=8037720011>.
- [13] "DHL drone," <https://www.dpdhl.com/en/media-relations/specials/dhl-parcelcopter.html>.
- [14] "DHL drone," <http://www.nazgoz.com/2016/06/fully-autonomous-delivery-drone-by-dhl.html>.
- [15] "UPS drone," <http://qualityfoodawards.com/news/google-promising-drones-will-be-able-to-deliver-food-to-drink-to-your-hand-within-two-years-13-01-2016/>.
- [16] "Google drone," <https://x.company/projects/wing/>.
- [17] "Google drone," <https://theredbanner.org/1030/the-spotlight/ups-drone-deliveries/>.
- [18] P. T. Biltgen, T. Ender, and D. N. Mavris, "Development of a collaborative capability-based tradeoff environment for complex system architectures," in *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 2006.
- [19] D. P. Raymer, *Aircraft design: a conceptual approach*, Fourth edition. American Institute of Aeronautics and Astronautics, Inc. (AIAA), 2006.
- [20] J. D. Mattingly, W. H. Heiser, and D. T. Pratt, *Aircraft engine design*, Second edition. American Institute of Aeronautics and Astronautics, Inc. (AIAA), 2002.
- [21] S. Defoort, M. Balesdent, P. Klotz, P. Schmollgruber, J. Morio, J. Hermetz, C. Blondeau, G. Carrier, and N. Bérend, "Multidisciplinary aerospace system design: Principles, issues and onera experience," *AerospaceLab*, vol. 4, pp. 1–15, 2012.
- [22] S. A. Brandt, "Small UAV design development and sizing," in *Handbook of Unmanned Aerial Vehicles*, K. Valavanis and G. Vachtsevanos, Eds., Springer, Dordrecht, 2015.
- [23] T. Nam, "A generalized sizing method for revolutionary concepts under probabilistic design constraints," PhD thesis, Georgia Institute of Technology, 2007.
- [24] J. Gundlach, *Designing unmanned aircraft systems: a comprehensive approach*. American Institute of Aeronautics and Astronautics, Inc. (AIAA), 2012.

- [25] B. Eksioglu, A. V. vural, and A. Reisman, "The vehicle routing problem: A taxonomic review," *Computers & Industrial Engineering*, vol. 57, no. 4, pp. 1472–1483, 2009.
- [26] Federal Aviation Administration (FAA). (2016). Fact sheet – small unmanned aircraft regulations (part 107).
- [27] Y. Choi, Y. Choi, S. I. Briceno, and D. N. Mavris, "An extended savings algorithm for UAS-based delivery systems," in *AIAA Scitech 2019 Forum (AIAA 2019-1796)*, San Diego, California, 2019.
- [28] Y. Mulgaonkar, M. Whitzer, B. Morgan, C. M. Kroninger, A. M. Harrington, and V. Kumar, "Power and weight considerations in small, agile quadrotors," *Proc. SPIE 9083, Micro-, Nanotechnology Sensors, Systems, and Applications VI*, June 4, 2014.
- [29] M. Mane, W. A. Crossley, and Nusawardhana, "System-of-systems inspired aircraft sizing and airline resource allocation via decomposition," *Journal of Aircraft*, vol. 44, no. 4, pp. 1222–1235, 2007.
- [30] C. Taylor and O. L. de Weck, "Coupled vehicle design and network flow optimization for air transportation systems," *Journal of Aircraft*, vol. 44, no. 5, pp. 1478–1486, 2007.
- [31] M. Mane and W. A. Crossley, "Allocation and design of aircraft for on-demand air transportation with uncertain operations," *Journal of Aircraft*, vol. 49, no. 1, pp. 141–150, 2012.
- [32] P. Govindaraju and W. A. Crossley, "Concurrent aircraft design and trip assignment under both design parameter and demand uncertainty," AIAA 2015-2545, 15th AIAA Aviation Technology, Integration, and Operations Conference (ATIO), Dallas, Texas, June 22-26, 2015.
- [33] M. L. Bociaga and W. A. Crossley, "A UAV design / sizing study from a system of systems perspective," AIAA 2013-0946, 51th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Grapevine, Texas, January 7-10, 2013.
- [34] E. Saliby, "Descriptive sampling: A better approach to Monte Carlo simulation," *The Journal of the Operational Research Society*, vol. 41, no. 12, pp. 1133–1142, 1990.
- [35] M. Tari and A. Dahmani, "Refined descriptive sampling: A better approach to Monte Carlo simulation," *Simulation Modelling Practice and Theory*, vol. 14, pp. 143–160, 2006.

- [36] C. Kelley, Ed., *Iterative methods for linear and nonlinear equation*. Society for Industrial and Applied Mathematics (SIAM), 1995.
- [37] A. Yeckel, A. Pandey, and J. J. Derby, “Fixed-point convergence of modular, steady-state heat transfer models coupling multiple scales and phenomena for melt–crystal growth,” *International Journal for Numerical Methods in Engineering*, vol. 67, pp. 1768–1789, 2006.
- [38] K. Bi, W. Zhang, C. Chi, and J. Zhang, “Reconfigurable control allocation of multi-surfaces aircraft based on improved fixed point iteration,” in *Foundations and Practical Applications of Cognitive Systems and Information Processing*, ser. Advances in Intelligent Systems and Computing 215, F. Sun, D. Hu, and H. Liu, Eds., Springer, Berlin, Heidelberg, 2014.
- [39] H.-J. Bungartz and M. Schäfer, Eds., *Fluid-Structure Interaction: Modelling, Simulation, Optimisation*, ser. Lecture Notes in Computational Science and Engineering 53. Springer-Verlag Berlin Heidelberg, 2006.
- [40] M. Müller, M. Woidt, M. Haupt, and P. Horst, “Challenges of fully-coupled high-fidelity ditching simulations,” *Aerospace*, vol. 6, no. 10, 2019.
- [41] B. Malone and W. H. Mason, “Multidisciplinary optimization in aircraft design using analytic technology models,” *Journal of Aircraft*, vol. 32, no. 2, pp. 431–438, 1995.
- [42] J. V. Lambers and A. C. Sumner, Eds., *Explorations in numerical analysis*. World Scientific Publishing Company, 2016.
- [43] G. Nemhauser and L. Wolsey, Eds., *Integer and combinatorial optimization*. John Wiley & Sons, 1999.
- [44] S. Onn and U. G. Rothblum, “Convex combinatorial optimization,” *Discrete & Computational Geometry*, no. 32, pp. 549–566, 2004.
- [45] T. Ibaraki, “Integer programming formulation of combinatorial optimization problems,” *Discrete Mathematics*, vol. 16, no. 1, pp. 39–52, 1976.
- [46] J. N. Hooker, *Integrated methods for optimization*, Second edition. Springer, 2012.
- [47] L. A. Wolsey, Ed., *Integer programming*. John Wiley & Sons, 1998.
- [48] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to automata theory, languages, and computation*. Pearson Education, 2001.

- [49] M. R. Garey and D. S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*, ser. Series of Books in the Mathematical Sciences. W. H. Freeman, 1979.
- [50] J. Kleinberg and E. Tardos, *Algorithm design*. Pearson, 2006.
- [51] S. A. Cook, “The complexity of theorem-proving procedures,” Proceedings of the third annual ACM symposium on Theory of computing, 1971, pp. 151–158.
- [52] R. M. Karp, “Reducibility among combinatorial problems,” *Complexity of Computer Computations*, pp. 85–103, 1972.
- [53] J. V. Leeuwen, *Algorithms and complexity*, ser. Handbook of Theoretical Computer Science, Vol. A. Elsevier Science, 1990.
- [54] G. Dantzig, R. Fulkerson, and S. Johnson, “Solution of a large-scale traveling-salesman problem,” *Operations Research*, vol. 2, pp. 393–410, 1954.
- [55] M. Jünger, G. Reinelt, and G. Rinaldi, *Network models*, M. Ball, T. Magnanti, B. Monma, and G. Nemhauser, Eds. Elsevier Science, 1995, vol. 7.
- [56] H. K.L., P. M., and R. G., *Traveling salesman problem*. Springer, Boston, MA, 2013.
- [57] R. Neapolitan and K. Naimipour, *Foundations of algorithms using C++ pseudocode*, third edition. Jones and Bartlett, 2004.
- [58] P. Toth and D. Vigo, Eds., *Vehicle routing: problems, methods, and applications*, second. SIAM, 2014.
- [59] G. B. Dantzig and J. H. Ramser, “The truck dispatching problem,” *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.
- [60] G. Clarke and J. R. Wright, “Scheduling of vehicles from a central depot to a number of delivery points,” *Operational Research*, vol. 12, no. 4, pp. 568–581, 1964.
- [61] R. Baldacci, M. Battarra, and D. Vigo, *Routing a heterogeneous fleet of vehicles, the vehicle routing problem: latest advances and new challenges*, B. Golden, S. Raghavan, and E. Wasil, Eds. Springer, 2008.
- [62] S. Martello and P. Toth, *Knapsack problems: algorithms and computer implementations*, ser. Wiley Interscience Series in Discrete Mathematics and Optimization. Wiley, New York, 1990.

- [63] B. Kallehauge, J. Larsen, O. B. Madsen, and M. M. Solomon, *Vehicle routing problem with time windows, column generation*, G. Desaulniers, J. Desrosiers, and M. M. Solomon, Eds. Springer, 2005.
- [64] H. Paessens, “The savings algorithm for the vehicle routing problem,” *European Journal of Operational Research*, vol. 34, no. 3, pp. 336–344, 1988.
- [65] R. Gopalan and K. T. Talluri, “Mathematical models in airline schedule planning: A survey,” *Annals of Operations Research*, vol. 76, pp. 155–185, 2003.
- [66] C. Barnhart, P. Belobaba, and A. R. Odoni, “Applications of operational research in the air transport industry,” *Transport Science*, vol. 37, no. 4, pp. 368–391, 2003.
- [67] J. Antes, L. Campen, U. Derigs, C. Titze, and G. D. Wolle, “Synopsis: A model-based decision support system for the evaluation of flight schedules for cargo airlines,” *Decision Support Systems*, vol. 22, no. 4, pp. 307–323, 1998.
- [68] D. Li, H. C. Huang, A. D. Morton, and E. P. Chew, “Simultaneous fleet assignment and cargo routing using benders decomposition,” *OR Spectrum*, vol. 28, no. 3, pp. 319–335, 2006.
- [69] S. Erdoğan and E. Miller-Hooks, “A green vehicle routing problem,” *Transportation Research Part E*, vol. 48, no. 1, pp. 100–114, 2012.
- [70] J. F. Bard, L. Huang, M. Dror, and P. Jaillet, “A branch and cut algorithm for the vrp with satellite facilities,” *IIE Transactions*, vol. 30, no. 9, pp. 821–834, 1998.
- [71] M. Schneider, A. Stenger, and D. Goeke, “The electric vehicle-routing problem with time windows and recharging stations,” *Transportation Science*, vol. 48, no. 4, pp. 500–520, 2014.
- [72] M. Schneider, A. Stenger, and J. Hof, “An adaptive VNS algorithm for vehicle routing problems with intermediate stops,” *OR Spectrum*, vol. 37, no. 2, pp. 353–387, 2015.
- [73] D. Goeke and M. Schneider, “Routing a mixed fleet of electric and conventional vehicles,” *European Journal of Operational Research*, vol. 245, no. 1, pp. 81–99, 2015.
- [74] G. Hiermann, J. Puchinger, S. Ropke, and R. F. Hartl, “The electric fleet size and mix vehicle routing problem with time windows and recharging stations,” *European Journal of Operational Research*, vol. 252, no. 3, pp. 995–1018, 2016.
- [75] B. Fleischmann, “The vehicle routing problem with multiple use of vehicles,” in. Working paper, Fachbereich Wirtschaftswissenschaften, Universität Hamburg, 1990.

- [76] D. Cattaruzza, N. Absi, and D. Feillet, “Vehicle routing problems with multiple trips,” *Operational Research*, vol. 14, no. 3, pp. 223–259, 2016.
- [77] C. Koc and I. Karaoglan, “A branch and cut algorithm for the vehicle routing problem with multiple use of vehicles,” in *Proceedings of the 41st international conference on computers & industrial engineering*, Los Angeles, USA, 2011, pp. 554–559.
- [78] E.-H. Aghezzaf, B. Raa, and H. V. Landeghem, “Modeling inventory routing problems in supply chains of high consumption products,” *European Journal of Operational Research*, vol. 169, no. 3, pp. 1048–1063, 2006.
- [79] N. Azi, M. Gendreau, and J.-Y. Potvin, “An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles,” *European Journal of Operational Research*, vol. 202, no. 3, pp. 756–763, 2010.
- [80] L. Lei, S. Liu, A. Ruszczyński, and S. Park, “On the integrated production, inventory, and distribution routing problem,” *IIE Transactions*, vol. 38, no. 11, pp. 955–970, 2006.
- [81] F. Hernandez, D. Feillet, R. Giroudeau, and O. Naud, “Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows,” *European Journal of Operational Research*, vol. 249, no. 2, pp. 551–559, 2016.
- [82] Y. Choi, B. Robertson, Y. Choi, and D. N. Mavris, “Formulation and implementation of a methodology of multi-trip vehicle routing for small UAS-based urban delivery,” *Journal of Aircraft*, 2019, Submitted for publication.
- [83] Y. Choi, H. Jimenez, and D. N. Mavris, “Two-layer obstacle collision avoidance with machine learning for more energy-efficient unmanned aircraft trajectories,” *Robotics and Autonomous Systems*, vol. 98, pp. 158–173, 2017.
- [84] S. M. LaValle, “Rapidly-exploring random trees: A new tool for path planning,” *Technical Report 98-11*, 1998.
- [85] O. Souissi, R. Benatitallah, D. Duvivier, A. Artiba, N. Belanger, and P. Feyzeau, “Path planning: A 2013 survey,” in *the 5th IESM Conference*, Rabat, Morocco, 2013.
- [86] J. Park, H. Oh, and M. Tahk, “UAV collision avoidance based on geometric approach,” in *SICE Annual Conference 2008*, 2008.
- [87] A. Chakravarthy and D. Ghose, “Obstacle avoidance in a dynamic environment: A collision cone approach,” *Systems, Man, and Cybernetics, Part A: Systems And Humans*, vol. 28, no. 5, 1998.

- [88] Y. Watanabe, A. Calise, and E. Johnson, "Vision-based obstacle avoidance for UAVs," in *AIAA Guidance, Navigation and Control Conference*, Hilton Head, South Carolina, 2007.
- [89] S. Tom, B. Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *European Control Conference*, Porto, Portugal, 2001.
- [90] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams," in *IEEE International Conference on Robotics and Automation (ICRA)*, Saint Paul, Minnesota, 2012.
- [91] H. C. Christmann and E. N. Johnson, "Modeling urban environments for communication-aware uav swarm path planning," in *AIAA Modeling and Simulation Technologies Conference*, Toronto, Ontario, Canada, 2014.
- [92] R. C. Arkin, "Path planning for a vision-based autonomous robot," in *Cambridge Symposium Intelligent Robotics Systems*, Cambridge, MA, United States, 1987.
- [93] T. Lozano-Pérez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the ACM*, vol. 22, no. 10, pp. 560–570, 1979.
- [94] G. Oh, Y. Kim, J. Ahn, and H.-L. Choi, "Market-based distributed task assignment of multiple unmanned aerial vehicles for cooperative timing mission," *Journal of Aircraft*, vol. 54, no. 6, pp. 2298–2310, 2017.
- [95] M. Sharir and A. Schorr, "On shortest paths in polyhedral spaces," *SIAM Journal on Computing*, vol. 15, pp. 193–215, 1986.
- [96] T. Asano, T. Asano, L. Guibas, J. Hershberger, and H. Imai, "Visibility of disjoint polygons," *Algorithmica*, vol. 1, pp. 49–63, 1986.
- [97] J. A. Storer and J. H. Reif, "Shortest paths in the plane with polygonal obstacles," *Journal of the ACM (JACM)*, vol. 41, no. 5, pp. 982–1012, 1994.
- [98] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, "Computational geometry," in, 3rd ed. Springer-Verlag, 2008, ch. 7.
- [99] F. Aurenhammer, "Voronoi diagrams - a survey of a fundamental geometric data structure," *ACM Comput. Surv.*, vol. 23, no. 3, pp. 345–405, Sep. 1991.
- [100] S. Garrido, L. Moreno, D. Blanco, and P. Jurewicz, "Path planning for mobile robot navigation using voronoi diagram and fast marching," *International Journal of Robotics and Automation (IJRA)*, vol. 2, no. 1, pp. 42–64, 1 2011.

- [101] V. P. Miloš Šeda, “Robot motion planning using generalized voronoi diagrams,” in *ISCGAV’08 Proceedings of the 8th conference on Signal processing, computational geometry and artificial vision*, Rhodes, Greece, 2008, pp. 215–220.
- [102] P. Bhattacharya and M. L. Gavrilova, “Density-based clustering based on topological properties of the data set,” in *Generalized Voronoi Diagram: A Geometry-Based Approach to Computational Intelligence*, ser. Studies in Computational Intelligence 158, M. L. Gavrilova, Ed., Springer-Verlag, 2008, pp. 197–214.
- [103] Y. Li, T. Dong, M. Bikdash, and Y. D. Song, “Path planning for unmanned vehicles using ant colony optimization on a dynamic voronoi diagram,” in *International Conference on Artificial Intelligence, ICAI*, Las Vegas, Nevada, United States, 2005.
- [104] Y. Eun and H. Bang, “Cooperative control of multiple unmanned aerial vehicles using the potential field theory,” *Journal of Aircraft*, vol. 43, no. 6, pp. 1805–1814, 2006.
- [105] J. Krozel and D. Andrisani II, “Navigation path planning for autonomous aircraft: Voronoi diagram approach,” *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 6, pp. 1152–1154, 1990.
- [106] I. A. McManus and R. A. Walker, “Multidisciplinary approach to intelligent unmanned-airborne-vehicles mission planning,” *Journal of Aircraft*, vol. 43, no. 2, pp. 318–335, 2006.
- [107] S. Upadhyay and A. Ratnoo, “Smooth path planing for unmanned aerial vehicles with airspace restrictions,” *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 7, pp. 1596–1612, 2017.
- [108] Y. Choi, “A framework for modeling and simulation of control, navigation, and surveillance for unmanned aircraft separation assurance,” PhD thesis, Georgia Institute of Technology, 2016.
- [109] J. Oyola, H. Arntzen, and D. L. Woodruff, “The stochastic vehicle routing problem, a literature review, part i: Models,” *EURO Journal on Transportation and Logistics*, vol. 7, pp. 193–221, 2018.
- [110] N. Lavars, “Amazon to begin testing new delivery drones in the US,” *GIZMAG*, 2015, <https://newatlas.com/amazon-new-delivery-drones-us-faa-approval/36957/>.
- [111] R. Kobeissi, “Flying high: Why enterprises are embracing drones,” *Orange Business Services*, 2017, <https://www.orange-business.com/en/blogs/>

connecting-technology/m2m/flying-high-why-enterprises-are-embracing-drones.

- [112] A. M. Kamal and A. Ramirez-Serrano, "Design methodology for hybrid (VTOL + fixed wing) unmanned aerial vehicles," *Aeronautics and Aerospace Open Access Journal*, vol. 2, no. 3, pp. 165–176, 2018.
- [113] M. Tyan, N. Nguyen, S. Kim, and J.-W. Lee, "Comprehensive preliminary sizing/resizing method for a fixed wing-VTOL electric UAV," *Aerospace Science and Technology*, vol. 71, pp. 30–41, 2017.
- [114] O. Gur and A. Rosen, "Optimizing electric propulsion systems for unmanned aerial vehicles," *Journal of Aircraft*, vol. 46, no. 4, pp. 1340–1353, 2009.
- [115] C. Ampatis and E. Papadopoulos, "Parametric design and optimization of multirotor aerial vehicles," in *2014 IEEE International Conference on Robotics & Automation (ICRA)*, Hong Kong, China, 2014.
- [116] D. Bershadsky, S. Haviland, and E. N. Johnson, "Electric multirotor propulsion system sizing for performance prediction and design optimization," in *AIAA Scitech 2016 Forum*, San Diego, California, 2016.
- [117] M. J. Stepaniak, F. van Graas, and M. U. de Haag, "Design of electric propulsion system for a quadrotor unmanned aerial vehicle," *Journal of Aircraft*, vol. 46, no. 3, pp. 1050–1058, 2009.
- [118] *MIL-HDBK-310, military handbook: Global climatic data for developing military products*.
- [119] S. Gudmundsson, *General aviation aircraft design: applied methods and procedures*. Butterworth-Heinemann, 2013.
- [120] J. G. Leishman, Ed., *Principles of helicopter aerodynamics*. Cambridge University Press, 2006.
- [121] J. Xu, Ed., *Design perspectives on delivery drones*. RAND Corporation, 2017.
- [122] J. John D. Anderson, *Aircraft performance and design*. McGraw-Hill, 1999.
- [123] C. Russell, J. Jung, G. Willink, and B. Glasner, "Wind tunnel and hover performance test results for multicopter UAS vehicles," in *72nd American Helicopter Society (AHS) International Annual Forum and Technology Display*, West Palm Beach, FL, 2016.

- [124] H. A. Kutty and P. Rajendran, “3D CFD simulation and experimental validation of small APC slow flyer propeller blade,” *Aerospace*, vol. 4, no. 10, 2017.
- [125] J. Brandt, R. Deters, G. Ananda, and M. Selig, “UIUC propeller database,” *University of Illinois at Urbana-Champaign*, <http://m-selig.ae.illinois.edu/props/propDB.html> (Accessed February 13, 2019).
- [126] “Performance data,” *APC propellers*, <https://www.apcprop.com/technical-information/performance-data/> (Accessed February 13, 2019).
- [127] D. L. Gabriel, J. Meyer, and F. du Plessis, “Brushless DC motor characterisation and selection for a fixed wing UAV,” in *IEEE Africon 2011 - The Falls Resort and Conference Centre*, Livingstone, Zambia, 2011.
- [128] E. K. C. IV, “Survey of small unmanned aerial vehicle electric propulsion system,” Master’s thesis, University of Dayton, 2007.
- [129] H. G. Brand, “Development and integration of an autonomous UAV into an urban security system,” Master’s thesis, The Potchefstroom campus of the North-West University, 2011.
- [130] Y. Choi, Y. Choi, S. Briceno, and D. N. Mavris, “Coverage path planning for a UAS imagery mission using column generation with a turn penalty,” in *The 2018 International Conference on Unmanned Aircraft Systems*, Dallas, TX, 2018, pp. 1109–1117.
- [131] Y. Choi, Y. Choi, S. Briceno, and D. N. Mavris, “Three-dimensional UAS trajectory optimization for remote sensing in an irregular terrain environment,” in *The 2018 International Conference on Unmanned Aircraft Systems*, Dallas, TX, 2018, pp. 1101–1108.
- [132] Y. Choi, Y. Choi, S. Briceno, and D. N. Mavris, “Energy-constrained multi-UAV coverage path planning for an aerial imagery mission using column generation,” *Journal of Intelligent and Robotic Systems*, 2019.
- [133] Y. Choi, M. Chen, Y. Choi, S. Briceno, and D. Mavris, “Multi-UAV trajectory optimization utilizing a NURBS-based terrain model for an aerial imaging mission,” *Journal of Intelligent and Robotic Systems*, 2019.
- [134] G. Laporte, M. Gendreau, J. Potvin, and F. Semet, “Classical and modern heuristics for the vehicle routing problem,” *International Transactions in Operational Research*, vol. 7, no. 4-5, pp. 285–300, 2000.

- [135] C. D. Franco and G. Buttazzo, "Coverage path planning for UAVs photogrammetry with energy and resolution constraints," *Journal of Intelligent & Robotic Systems*, vol. 83, pp. 445–462, 2016.
- [136] W. H. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *Proceedings of the 2001 IEEE International Conference on Robotics & Automation*, Seoul, Korea, 2001.
- [137] I. KARA, "Arc based integer programming formulations for distance constrained vehicle routing problem," in *LINDI 2011 - 3rd IEEE International Symposium on Logistics and Industrial Informatics*, Budapest, Hungary, 2011.
- [138] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programmings," *Operations Research*, vol. 8, pp. 101–111, 1960.
- [139] M. Jünger, T. Liebling, D. Naddef, G. Nemhauser, W. Pulleyblank, G. Reinelt, G. Rinaldi, and L. Wolsey, Eds., *50 years of integer programming 1958-2008: from the early years to the state-of-the-art*. Springer, 2010.
- [140] G. Desaulniers, J. Desrosiers, and M. M. Solomon, Eds., *Column generation*. Springer, 2005.
- [141] P. C. Gilmore and R. E. Gomory, "A linear programming approach to the cutting-stock problem," *Operations Research*, vol. 9, no. 6, pp. 849–859, 1961.
- [142] P. C. Gilmore and R. E. Gomory, "A linear programming approach to the cutting-stock problem-part 2," *Operations Research*, vol. 11, no. 6, pp. 863–888, 1963.
- [143] M. Desrochers and F. Soumis, "A column generation approach to the urban transit crew scheduling problem," *Transportation Science*, vol. 23, no. 1, pp. 1–13, 1989.
- [144] M. Desrochers, J. Desposiers, and M. Solomon, "A new optimization algorithm for the vehicle routing problem with time windows," *Operations Research*, vol. 40, no. 2, pp. 342–354, 1992.
- [145] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance, "Branch-and-price: Column generationn for solving huge integer probrams," *INFORMS*, vol. 46, no. 3, pp. 316–329, 1998.
- [146] F. Mufalli, R. Batta, and R. Nagi, "Simultaneous sensor selection and routing of unmanned aerial vehicles for complex mission plans," *Computer & Operations Research*, vol. 39, pp. 2787–2799, 2012.
- [147] J. Zillies, S. Westphal, D. Thakur, V. Kumar, G. Pappas, and D. Scheidt, "A column generation approach for optimized routing and coordination of a UAV fleet,"

in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, EPFL, Lausanne, Switzerland, 2016.

- [148] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen, “An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problem,” *Networks*, vol. 44, no. 3, pp. 216–229, 2004.
- [149] Y. Choi, Y. Choi, S. Briceno, and D. N. Mavris, “Multi-UAS path-planning for a large-scale disjoint disaster management,” in *The 2019 International Conference on Unmanned Aircraft Systems*, Accepted for publication, 2019.
- [150] V. G. Ambrosia, S. S. Wegener, J. A. Brass, and S. M. Schoenung, “The UAV western states fire mission: Concepts, plans and developmental advancements,” in *AIAA 3rd ”Unmanned Unlimited” Technical Conference, Workshop and Exhibit*, Chicago, IL, 2004.
- [151] S. S. Wegener, S. M. Schoenung, J. Totah, D. Sullivan, J. Frank, F. Enomoto, C. Frost, and C. Theodore, “UAV autonomous operations for airborne science missions,” in *AIAA 3rd ”Unmanned Unlimited” Technical Conference, Workshop and Exhibit*, Chicago, IL, 2004.
- [152] D. W. Casbeer, D. B. Kingston, R. W. Beard, and T. W. McLain, “Cooperative forest fire surveillance using a team of small unmanned air vehicles,” *International Journal of Systems Science*, vol. 37, no. 6, pp. 351–360, 2006.
- [153] C. Phan and H. H. Liu, “A cooperative UAV/UGV platform for wildfire detection and fighting,” in *2008 Asia Simulation Conference - 7th International Conference on System Simulation and Scientific Computing*, Beijing, China, 2008.
- [154] H. X. Pham, H. M. La, D. Feil-Seifer, and M. Deans, “A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 2017.
- [155] R. Bailon-Ruiz, A. Bit-Monnot, and S. Lacroix, “Planning to monitor wildfires with a fleet of UAVs,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Madrid, Spain, 2018.
- [156] J. I. Vasquez-Gomez, J.-C. Herrera-Lozada, and M. Olguin-Carbajal, “Coverage path planning for surveying disjoint areas,” in *The 2018 International Conference on Unmanned Aircraft Systems*, Dallas, TX, 2018, pp. 899–904.
- [157] J. Valente, D. Sanz, J. D. Cerro, A. Barrientos, and M. A. de Frutos, “Near-optimal coverage trajectories for image mosaicing using a mini quad-rotor over irregular-shaped fields,” *Precision Agric*, vol. 14, pp. 115–132, 2013.

- [158] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” *Robotics and Autonomous Systems*, vol. 61, pp. 1258–1276, 2013.
- [159] A. Khan, I. Noreen, and Z. Habib, “On complete coverage path planning algorithms for non-holonomic mobile robots: Survey and challenges,” *Journal of Information Science and Engineering*, vol. 33, pp. 101–121, 2017.
- [160] L. E. Dubins, “On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents,” *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.
- [161] H. H. Johnson, “An application of the maximum principle to the geometry of plane curves,” *Proceedings of the American Mathematical Society*, vol. 44, no. 2, pp. 432–435, 1974.
- [162] J. Boissonnat, A. Cérézo, and J. Leblond, “Shortest paths of bounded curvature in the plane,” *Journal of Intelligent and Robotic Systems*, vol. 11, pp. 5–20, 1994.
- [163] B. L. Golden, T. L. Magnanti, and H. Q. Nguyen, “Implementing vehicle routing algorithms,” *Networks*, vol. 7, pp. 113–148, 1977.
- [164] R. V. Kulkarni and P. R. Bhave, “Integer programming formulations of vehicle routing problems,” *European Journal of Operational Research*, vol. 20, pp. 58–67, 1985.
- [165] A. Nedjati, G. Izbirak, B. Vizvari, and J. Arkat, “Complete coverage path planning for a multi-UAV response system in post-earthquake assessment,” *Robotics*, vol. 26, no. 5, 2016.
- [166] J. R. Montoya-Torres, J. L. Franco, S. N. Isaza, H. F. Jiménez, and N. Herazo-Padilla, “A literature review on the vehicle routing problem with multiple depots,” *Computers and Industrial Engineering*, vol. 79, pp. 115–129, 2015.
- [167] S. Salhi, A. Imran, and N. A. Wassan, “The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation,” *Computers and Operations Research*, vol. 52, pp. 315–325, 2014.
- [168] M. Valdovinos, J. Specht, and J. Zeunik. (2016). Law enforcement & unmanned aircraft systems (UAS): Guidelines to enhance community trust, Washington, DC: Office of Community Oriented Policing Services.
- [169] National League of Cities (NLC), “Cities and drones: What cities need to know about unmanned aerial vehicles (UAVs),” Tech. Rep., 2016.

- [170] Federal Aviation Administration (FAA). (2016). Summary of small unmanned aircraft rule (part 107).